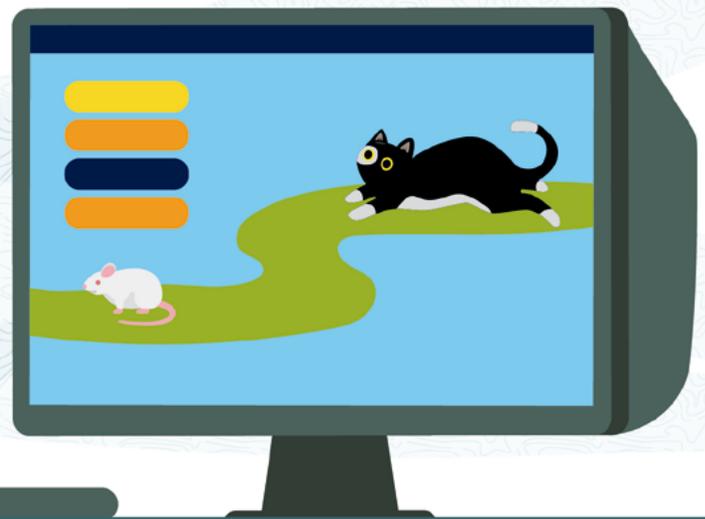
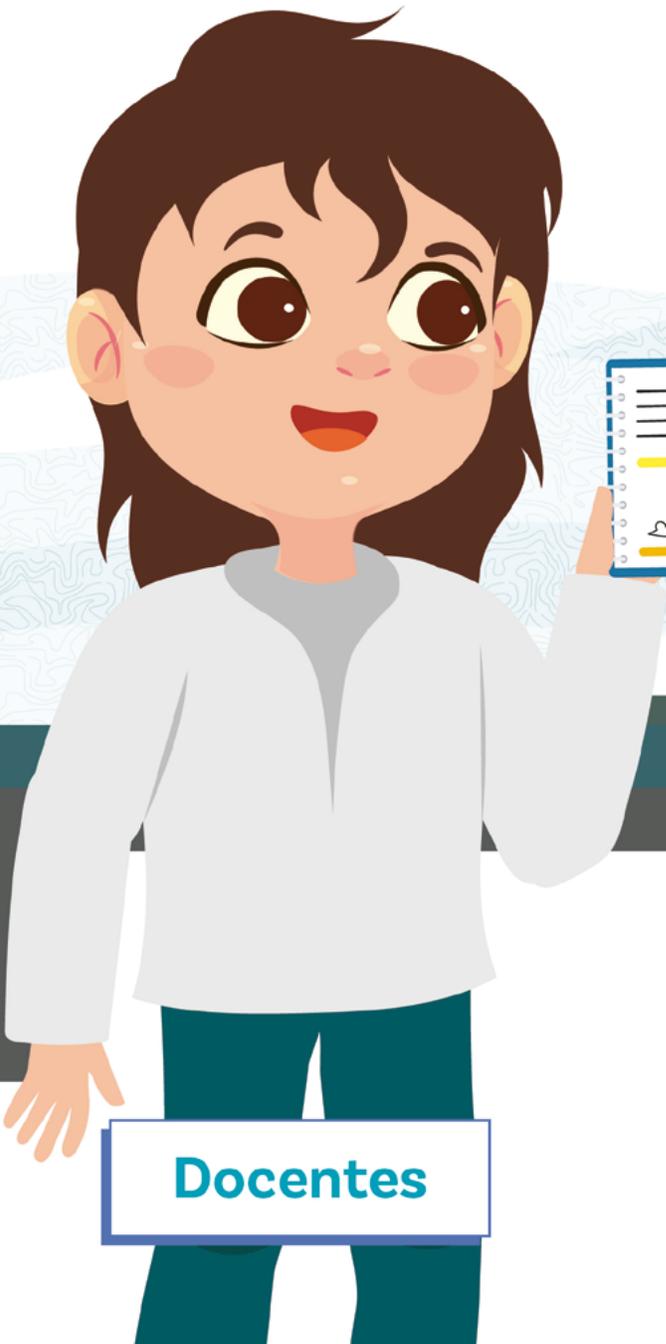


A que te cojo ratón

Grado 4°

Guía 4



Docentes

Apoya:



A que te cojo ratón

**Grado 4°**

**Guía 4**



**Docentes**



**MINISTERIO DE TECNOLOGÍAS  
DE LA INFORMACIÓN Y LAS  
COMUNICACIONES**

Julián Molina Gómez  
**Ministro TIC**

Luis Eduardo Aguiar Delgadillo  
**Viceministro (e) de Conectividad**

Yeimi Carina Murcia Yela  
**Viceministra de Transformación Digital**

Óscar Alexander Ballen Cifuentes  
**Director (e) de Apropiación de TIC**

Alejandro Guzmán  
**Jefe de la Oficina Asesora de Prensa**

**Equipo Técnico**  
Lady Diana Mojica Bautista  
Cristhiam Fernando Jácome Jiménez  
Ricardo Cañón Moreno

**Consultora experta**  
Heidy Esperanza Gordillo Bogota

**BRITISH COUNCIL**

Felipe Villar Stein  
**Director de país**

Laura Barragán Montaña  
**Directora de programas de Educación,  
Inglés y Artes**

Marianella Ortiz Montes  
**Jefe de Colegios**

David Vallejo Acuña  
**Jefe de Implementación  
Colombia Programa**

**Equipo operativo**  
Juanita Camila Ruiz Díaz  
Bárbara De Castro Nieto  
Alexandra Ruiz Correa  
Dayra Maritza Paz Calderón  
Saúl F. Torres  
Óscar Daniel Barrios Díaz  
César Augusto Herrera Lozano  
Paula Álvarez Peña

**Equipo técnico**  
Alejandro Espinal Duque  
Ana Lorena Molina Castro  
Vanesa Abad Rendón  
Raisa Marcela Ortiz Cardona  
Juan Camilo Londoño Estrada

**Edición y coautoría versiones finales**  
Alejandro Espinal Duque  
Ana Lorena Molina Castro  
Vanesa Abad Rendón  
Raisa Marcela Ortiz Cardona

**Edición**  
Juanita Camila Ruiz Díaz  
Alexandra Ruiz Correa

**British Computer Society –  
Consultoría internacional**

Niel McLean  
**Jefe de Educación**

Julia Adamson  
**Directora Ejecutiva de Educación**

Claire Williams  
**Coordinadora de Alianzas**

**Asociación de facultades de  
ingeniería - ACOFI**

**Edición general**  
Mauricio Duque Escobar

**Coordinación pedagógica**  
Margarita Gómez Sarmiento  
Mariana Arboleda Flórez  
Rafael Amador Rodríguez

**Coordinación de producción**  
Harry Luque Camargo

**Asesoría estrategia equidad**  
Paola González Valcárcel

**Asesoría primera infancia**  
Juana Carrizosa Umaña

**Autoría**  
Arlet Orozco Marbello  
Harry Luque Camargo  
Isabella Estrada Reyes  
Lucio Chávez Mariño  
Margarita Gómez Sarmiento  
Mariana Arboleda Flórez  
Mauricio Duque Escobar  
Paola González Valcárcel  
Rafael Amador Rodríguez  
Rocío Cardona Gómez  
Saray Piñerez Zambrano  
Yimzay Molina Ramos

**PUNTOAPARTE EDITORES**

Diseño, diagramación, ilustración,  
y revisión de estilo

Impreso por Panamericana Formas e  
Impresos S.A., Colombia

Material producido para Colombia  
Programa, en el marco del convenio  
1247 de 2023 entre el Ministerio de  
Tecnologías de la Información y las  
Comunicaciones y el British Council

Esta obra se encuentra bajo una  
Licencia Creative Commons  
Atribución-No Comercial  
4.0 Internacional. [https://  
creativecommons.org/licenses/  
by-nc/4.0/](https://creativecommons.org/licenses/by-nc/4.0/)



“Esta guía corresponde a una  
versión preliminar en proceso  
de revisión y ajuste. La versión  
final actualizada estará  
disponible en formato digital  
y puede incluir modificaciones  
respecto a esta edición”

# Prólogo

Estimados educadores, estudiantes y comunidad educativa:

En el Ministerio de Tecnologías de la Información y las Comunicaciones, creemos que la tecnología es una herramienta poderosa para incluir y transformar, mejorando la vida de todos los colombianos. Nos guía una visión de tecnología al servicio de la humanidad, ubicando siempre a las personas en el centro de la educación técnica.

Sabemos que no habrá progreso real si no garantizamos que los avances tecnológicos beneficien a todos, sin dejar a nadie atrás. Por eso, nos hemos propuesto una meta ambiciosa: formar a un millón de personas en habilidades que les permitan no solo adaptarse al futuro, sino construirlo con sus propias manos. Hoy damos un paso fundamental hacia este objetivo con la presentación de las guías de pensamiento computacional, un recurso diseñado para llevar a las aulas herramientas que fomenten la creatividad, el pensamiento crítico y la resolución de problemas.

Estas guías no son solo materiales educativos; son una invitación a imaginar, cuestionar y crear. En un mundo cada vez más impulsado por la inteligencia artificial, desarrollar habilidades como el pensamiento computacional se convierte en la base, en el primer acercamiento para que las y los ciudadanos aprendan a programar y solucionar problemas de forma lógica y estructurada.

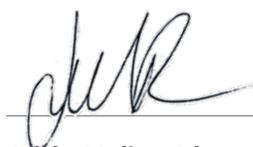
Estas guías han sido diseñadas pensando en cada región del país, con actividades accesibles que se adaptan a diferentes contextos, incluyendo aquellos con limitaciones tecnológicas. Esta es una apuesta por la equidad, por cerrar las brechas y asegurar que nadie se quede atrás en la revolución digital. Quiero destacar, además, que son el resultado de un esfuerzo colectivo:

más de 2.000 docentes colaboraron en su elaboración, compartiendo sus ideas y experiencias para que este material realmente se ajuste a las necesidades de nuestras aulas. Además, con el apoyo del British Council y su red de expertos internacionales, hemos integrado prácticas globales de excelencia adaptadas a nuestra realidad nacional.

Hoy presentamos un recurso innovador y de alta calidad, diseñado en línea con las orientaciones curriculares del Ministerio de Educación Nacional. Cada página de estas guías invita a transformar las aulas en espacios participativos, creativos y, sobre todo, en ambientes donde las y los estudiantes puedan desafiar estereotipos y explorar nuevas formas de pensar.

Trabajemos juntos para garantizar que cada estudiante, sin importar dónde se encuentre, tenga acceso a las herramientas necesarias para imaginar y construir un futuro en el que todos seamos protagonistas del cambio. Porque la tecnología debe ser un instrumento de justicia social, y estamos comprometidos a que las herramientas digitales ayuden a cerrar brechas sociales y económicas, garantizando oportunidades para todos.

Con estas guías, reafirmamos nuestro compromiso con la democratización de las tecnologías y el desarrollo rural, porque creemos en el potencial de cada región y en la capacidad de nuestras comunidades para liderar el cambio.



**Julián Molina Gómez**  
Ministro de Tecnologías de la  
Información y las Comunicaciones  
**Gobierno de Colombia**



## Guía de íconos



Algoritmos, patrones, abstracción y descomposición



Lógica, programación y depuración

## Aprendizajes de la guía

Con las actividades de esta guía se espera que sus estudiantes puedan:



Identificar y aplicar bucles o ciclos para estructurar secuencias repetidas de instrucciones o que deben ejecutarse según una o dos condiciones lógicas.



Reconocer y utilizar bucles o ciclos para describir secuencias repetidas de instrucciones o que deben ejecutarse según una o dos condiciones lógicas.



Modificar programas para incluir acciones repetitivas a través de bucles.

## Resumen de la guía

En esta guía, sus estudiantes tienen la oportunidad de avanzar en el uso de bucles o ciclos para resumir y simplificar programas y algoritmos. Además, avanzan en la creación de programas en *Scratch* utilizando diferentes tipos de bucles y variables.

## Resumen de las sesiones

## Sesión 1

El objetivo principal de la sesión es retomar el uso y la comprensión respecto a los bucles en la programación. Sus estudiantes exploran nuevas situaciones en las que pueden aplicarse, esta vez en un contexto básico de videojuegos.

## Sesión 2

En esta sesión se continúan reforzando los conocimientos sobre bucles. Sus estudiantes realizan ejercicios prácticos que incluyen dibujar un cuadrado y luego un hexágono. A través de estos ejercicios, sus estudiantes aplican el concepto de bucles para automatizar movimientos del lápiz, mientras reflexionan sobre la optimización del código y proponen mejoras.

## Sesión 3

En esta sesión, se profundiza en el uso de bucles en la programación, específicamente en el entorno de *Scratch*. Sus estudiantes aprenden cómo los bucles pueden optimizar su trabajo al automatizar tareas repetitivas y reducir la cantidad de código necesario.

## Aprendizajes de la guía



Crear, analizar y optimizar un programa en un lenguaje de bloques utilizando secuencias de instrucciones.

### Sesión 4

Esta sesión, sus estudiantes desarrollan habilidades para diseñar algoritmos más complejos, optimizando sus programas mediante el uso de bucles condicionales. Es decir, aquellos que dependen de una condición para permanecer en el bucle. Además, se espera que sus estudiantes adquieran una comprensión más sólida de cómo estos conceptos pueden utilizarse para resolver una variedad de problemas y crear proyectos más interactivos y dinámicos.

### Sesión 5

Esta sesión consolida lo aprendido en las sesiones anteriores y presenta ejercicios prácticos para utilizar diferentes tipos de bucles, incluyendo aquellos que usan condiciones para detenerse.

### Sesión 6

Esta sesión se dedica a recapitular lo aprendido y a la realización de una actividad de evaluación.

## Evaluación

A lo largo de las sesiones se encuentran oportunidades para evaluar formativamente, en las actividades de práctica y en los momentos de discusión con toda la clase.

Para cada sesión hay un conjunto de aprendizajes esperados que se podrán verificar a partir de los diálogos y productos desarrollados en la clase en el entorno de programación *Scratch*. Las actividades incluyen trabajo individual, en parejas y en grupos, lo que le permitirá alcanzar diferentes niveles de desarrollo de los aprendizajes y potenciarlos a través de la interacción entre sus estudiantes. También se incluyen actividades de cierre que le permitirá reforzar los aprendizajes e identificar dudas sobre las que pueda trabajar.

Además, la última sesión de la guía incluye una evaluación formal que se podrá adaptar para ver el progreso de cada estudiante. Esta se encuentra en el Anexo 6.1 y las soluciones a esta evaluación se presentan en el Anexo 6.2.



## Preparación de materiales y actividades

Cada sesión indica los materiales requeridos. Se recomienda prepararlos previamente para que su distribución y recolección tome el menor tiempo posible.

Se recomienda igualmente realizar las actividades propuestas antes de trabajarlas con sus estudiantes. Este es un factor clave en la planeación.

En esta guía se trabaja con Scratch, para lo cual se requiere de computadores o dispositivos móviles como celulares o tabletas y la aplicación local previamente instalada, o con conexión a Internet, para trabajar directamente con la versión en línea. Si no dispone de los elementos tecnológicos indicados, puede usar una versión de Scratch imprimible en 2D o 3D y trabajar con la programación en bloques, de forma desconectada:

<https://www.scratchjrtactile.org/start>



## Conexión con otras áreas

Esta guía se concentra en el desarrollo de habilidades propias del pensamiento computacional, sin embargo, aborda temas relacionados con otras áreas como:

### Geometría

- Dibujo de figuras simples y uso de ángulos.

### Matemáticas

- Uso de variables, suma, resta y comparación de mayor que, menor que.

# Sesión

# 1

## Aprendizajes esperados

Al final de esta sesión se espera que sus estudiantes puedan:

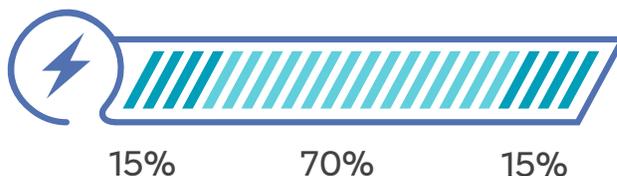


Explicar correctamente el concepto de bucle.



Modificar programas para incluir acciones repetitivas a través de bucles.

## Duración sugerida



## Material para la clase

- **Material por grupo:**  
Tarjetas recortadas del Anexo 1.1 y 1.2



**Enlace**

¿Qué son los ciclos en programación?

**Lo que sabemos,  
lo que debemos saber**

Esta sección corresponde al 15% de avance de la sesión

Antes de empezar, indique a sus estudiantes que con esta guía seguirán reforzando lo aprendido con relación a los bucles. Dígales que para empezar va a mostrarles un video en el que una persona le cuenta a otra un problema que tiene al crear un programa y que deben prestar atención a la situación planteada para que puedan plantear posibles soluciones. Luego, acceda al video *¿Qué son los ciclos en programación?*, accesible desde el QR y enlace provistos. Muestre hasta el segundo 24, donde el joven explica su problema. Pause el video y permita que sus estudiantes piensen en qué le podrían sugerir como solución.

Permita que sus estudiantes discutan sus ideas en parejas y luego escuche a unas dos o tres personas plantear posibles soluciones. Después, muéstrelas el resto del video para que comparen las soluciones que compartieron con las que se presentan allí. Aproveche el resto del video para reforzar los conocimientos de sus estudiantes frente a los bucles. Aclare que, al hablar de bucles, también se pueden utilizar las palabras repetición o ciclo.

Como ejercicio de activación de los conocimientos, realice preguntas tales como:



*En su vida diaria, ¿cuándo realizan procesos repetitivos?*

*¿En la naturaleza han visto algún proceso repetitivo?*

*¿Cuáles?*





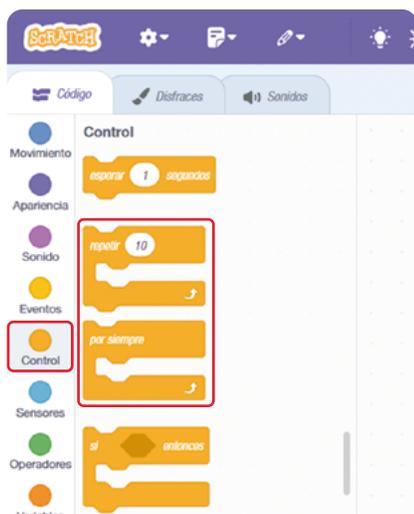
## Adaptación

1. En esta sesión y las siguientes se utilizará el entorno de programación *Scratch*, por ello, si tiene estudiantes con discapacidad visual asegúrese de contar con un lector de pantalla como NVDA o JAWS, y de utilizar estrategias como “Pares amigos”, donde una compañera o compañero pueda apoyar a su estudiante con discapacidad visual leyéndole las instrucciones y guiándole al navegar en la interfaz del programa.
2. Si no le es posible mostrar el video sugerido, puede reemplazar esta actividad por una en la que pida a algunas personas representar actividades y repetirlas un número de veces o hasta que indique. Ej. Salta 3 veces, camina hasta que llegues a la puerta, dibuja flores en el tablero hasta que completes 3, etc. Luego, utilice estas acciones como oportunidad para recordarles a sus estudiantes el concepto de bucle.

Explíqueles a sus estudiantes que para resolver problemas utilizando bucles se requieren habilidades de pensamiento lógico. Sin embargo, practicar con diversos ejercicios les ayudará a desarrollarlas y les permitirá encontrar diferentes formas de abordar un problema, así como explorar cómo pueden usar los bucles de manera efectiva en sus proyectos.

Recuerde la información que se presentó en el video y haga énfasis en que, en programación, a menudo es necesario ejecutar un conjunto de instrucciones múltiples veces. Para evitar la repetición manual de esos bloques de instrucciones, se agrupan dentro de una estructura de bucle, que se encarga de repetirlas automáticamente.

**Figura 1.** Ubicación de los bloques **Por siempre** y **Repetir** en la categoría de bloques de **Control** de Scratch



Pida que mencionen los diferentes tipos de bucles que han utilizado en Scratch y expliquen sus diferencias. La *Figura 1* presenta los bloques de bucles que han presentado y utilizado en guías anteriores.

Explique que, además, en Scratch existe otro tipo de bucle que es el bloque **Repetir hasta que (condición)** y aclare que aprenderán a usarlo posteriormente mientras avanzan desarrollando actividades de las diferentes sesiones de la guía.

## Manos a la obra



Esta sección corresponde al 85% de avance de la sesión



La actividad que se plantea en esta actividad les ayudará a sus estudiantes a recordar y practicar el uso de bloques **repetir** para incluir bucles en sus programas. También se les solicitará empezar a documentar programas, creando un registro de sus observaciones sobre lo que hacen los diferentes bloques de código. Para este fin se han creado fichas de anotaciones que se presentan como parte de los anexos.

Tome una de las fichas de anotaciones recortadas de los *Anexos 1.1* y *1.2* y explique que usted les irá entregando cada una de estas fichas en diferentes momentos de la clase para que escriban por un minuto lo que observan en el código. Pídales que vayan pegando estas fichas en su cuaderno para que, al finalizar la guía, tengan un resumen de su trabajo.

Si le es posible, indique a sus estudiantes que trabajen de forma individual. Pídales que ingresen a Scratch y vayan siguiendo su demostración e instrucciones. Si esto no es posible, divida la clase en parejas y asigne los roles piloto(a) y navegante. A las personas con el rol de pilotos permítale ser quienes codifican y a las que tengan el rol de navegantes pídale que vayan diciéndole a sus compañeras o compañeros lo que deben hacer según el paso que usted haya demostrado. Indique que, cuando se les avise, deberán cambiar roles para que ambas personas en cada pareja puedan manipular el dispositivo para programar.

## Anexo

## Anexo 1.2

Fichas anotaciones 2	Fichas anotaciones 2
Fichas anotaciones 2	Fichas anotaciones 2
Fichas anotaciones 2	Fichas anotaciones 2
Fichas anotaciones 2	Fichas anotaciones 2

Cuando todo el grupo tenga claro su rol, indique que van a crear un programa en el que un balón se mueva automáticamente en la pantalla. Comience creando un pequeño programa en el que un balón avance 10 pasos cada vez que se presione la bandera verde. Para ello, siga estos pasos:

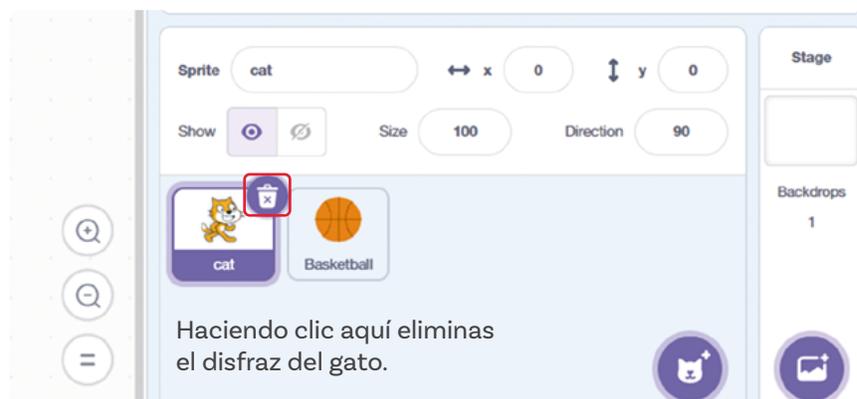
- 1 Agregue el disfraz de balón. Acceda a la sección de disfraces, como se ve en la *Figura 2* y seleccione la opción “Deportes”. A continuación, elija la imagen de un balón. Puede utilizar una imagen de baloncesto, pero también puede permitir que sus estudiantes seleccionen el balón del deporte que más les guste.

**Figura 2.** Vista del botón para ingresar a los disfraces



- 2 Al seleccionar un balón, aparecerán dos objetos en pantalla. Proceda a eliminar el objeto del gato, de modo que solo quede el disfraz del balón seleccionado, como se ve en la *Figura 3*.

**Figura 3.** Eliminar disfraz del gato



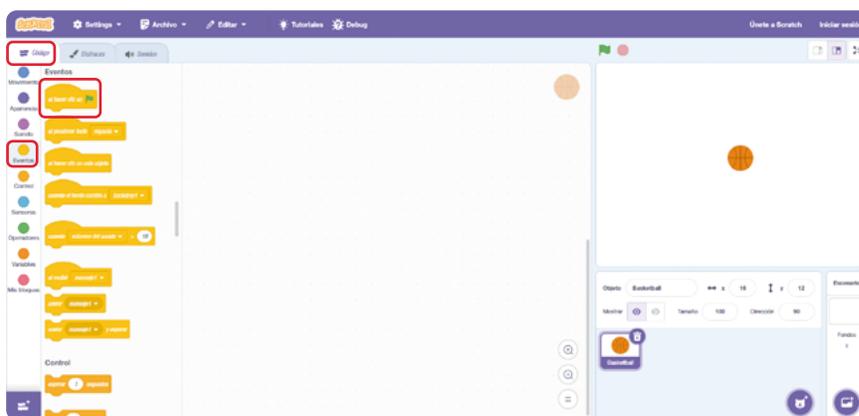
**Figura 6.** Programa para que rebote la pelota y se devuelva si toca un borde del escenario



Una vez que haya eliminado correctamente el objeto del gato y solo quede el objeto de la pelota (por ejemplo, el de baloncesto), continúe con la estructura del programa:

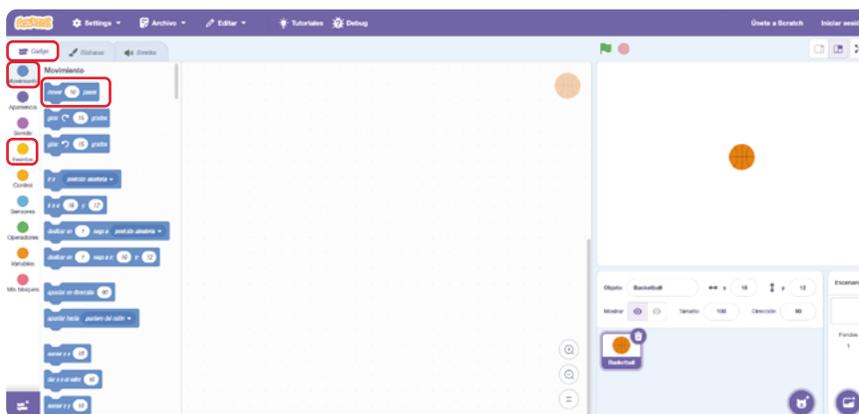
- 1 Diríjase a la pestaña **Código** y, en la categoría **Eventos**, seleccione el bloque **Al hacer clic en (bandera verde)**, como se ve en la *Figura 4*.

**Figura 4.** Selección del bloque iniciador de eventos **Al hacer clic en bandera verde**.



- 2 En la opción "Movimiento", seleccione el bloque **Mover (10) pasos**, como se ve en la *Figura 5*.

**Figura 5.** Selección del bloque **Mover 10 pasos**



- 3 Agregue el bloque **Si toca un borde, rebotar**, que se encuentra también en la categoría de bloques de movimiento. Asegúrese de que el programa que ha creado replica el que se muestra en la *Figura 6*.

**Figura 7.** Programa para mover de forma indefinida la pelota, rebotando al tocar el borde del escenario. Se usa el bucle **Por siempre**.



Dé a sus estudiantes un momento para que recreen el programa en sus computadores. Pase por los puestos y ayúdeles a identificar errores.

Una vez armado el programa, haga clic en la bandera verde una vez para que el código se ejecute, y pregunte a sus estudiantes qué pasa.

Entregue a cada estudiante una ficha recortada del Anexo 1.1.

Luego, pídeles hacer clic en la bandera verde cinco veces. Pida que presten atención a lo que ocurre, y que escriban sus observaciones personales en la ficha “Anotaciones 1” que les entregó.

Haga preguntas como:



*¿Qué pasa si quiero que la pelota se mueva varias veces?  
¿Cómo podría lograr que la pelota se mueva sin tener que presionar tantas veces la bandera verde?*

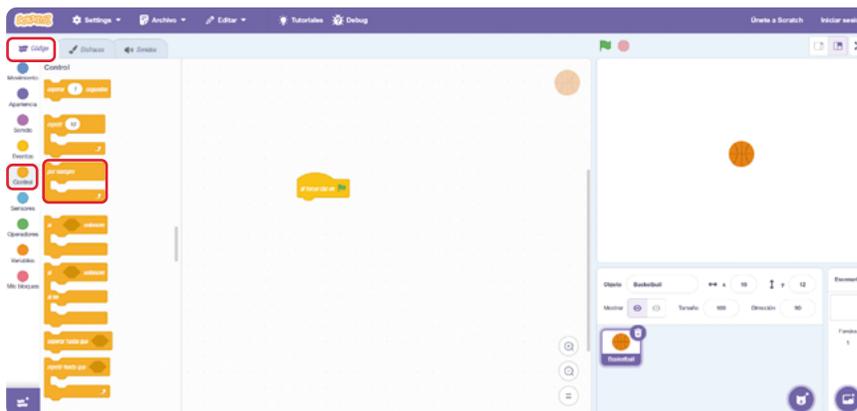
Permita que sus estudiantes predigan el bloque que utilizarán para lograrlo.

Ahora recuérdelos que pueden solucionar esto usando bucles, y que usted les demostrará cómo. Elimine todos los bloques del programa que realizó en el paso anterior y siga estas instrucciones:

- 1 Diríjase nuevamente a la pestaña **Código** y, en la categoría **Eventos**, seleccione el bloque **Al hacer clic en (bandera verde)**, como se ve en la Figura 4.

- 2 En la categoría **Control**, seleccione el bloque **Por siempre** y arrástrelo al programa, como se ve en la *Figura 8*.

**Figura 8.** Selección del bloque Por siempre



- 3 Regrese a la categoría **Movimiento** y agregue los bloques **Mover (10) pasos** y **Si toca un borde, rebotar**, de manera que el programa quede como se muestra en la *Figura 7*.

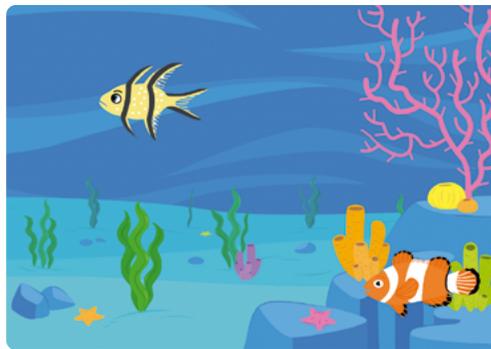
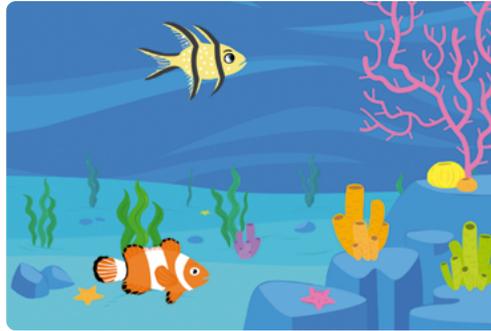
Con el programa listo, invite a sus estudiantes a hacer clic en la bandera y observar lo que sucede. Entregue a cada persona una ficha recortada del Anexo 1.2 y pídale que escriban sus observaciones allí.

Pida a sus estudiantes comparar sus anotaciones sobre el código de la *Figura 6* y el de la *Figura 7*, y discutir en parejas preguntas tales como:



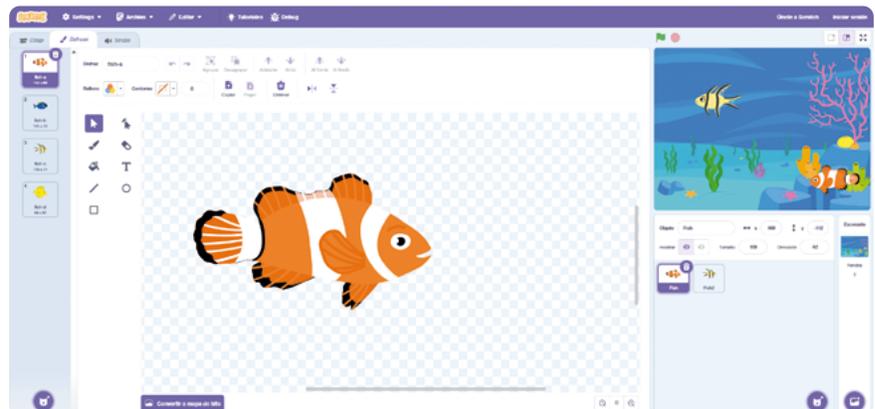
*¿Qué diferencias encontraron entre los dos programas?  
¿Cuál programa es más eficiente? ¿Por qué?  
¿Qué ventajas tiene utilizar bucles en la programación?*

Proponga a sus estudiantes utilizar lo repasado sobre bucles para solucionar un reto. Díales que su misión consistirá en diseñar un acuario virtual que puedan utilizar después como ayuda visual para contar un breve cuento. Sugiera, por tanto, que trabajen en la animación de una escena marina en la que aparezca un fondo de océano con dos peces que se muevan de un lado a otro, como se muestra en la *Figura 9*.

**Figuras 9.** Vista del reto a desarrollar a continuación

Para hacerlo, deben reutilizar el código que acaban de crear.

Invite a quienes tengan dificultades para empezar a levantarse silenciosamente y observar lo que hacen otras personas de la clase, esto puede servirles para desbloquearse y continuar. No obstante, ofrezca apoyo a quienes sigan necesitándolo. Por ejemplo, puede recordarles cómo cambiar el disfraz de los peces, como se ve en la *Figura 10*.

**Figura 10.** Selección de un disfraz diferente para el objeto Fish (pez)

A quienes terminen rápido el programa puede pedirles, además, que hagan modificaciones al código para que los peces se muevan a velocidades diferentes.

## Glosario



**Bucle:** estructura en programación que permite repetir un conjunto de instrucciones varias veces, facilitando la automatización de tareas repetitivas y la eficiencia en el código.

## Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

Cuando sus estudiantes hayan finalizado el programa, relacione lo aprendido en la sesión con la vida cotidiana. Genere una conversación preguntando:



*¿Para qué utilizaron los bucles en su programa?*  
*¿Qué otras acciones podrían querer que se realicen por siempre dentro de un programa?*  
*¿Pueden mencionar otros ejemplos de aplicaciones o elementos que utilizan bucles indefinidos como el que se usó hoy?*

Apoye la reflexión mencionando ejemplos de dispositivos como: un reloj que siempre debe estar mostrando la hora, una alarma contra incendios que siempre debe estar leyendo la temperatura y comparándola con la temperatura máxima permitida a fin de activarse automáticamente.

Indique que todo lo aprendido en estas sesiones corresponde a las bases del funcionamiento de muchas aplicaciones del mundo real. Recuerde registrar en un gráfico de anclaje lo que indican sus estudiantes que han aprendido. El siguiente es un posible ejemplo del gráfico que podría hacer. Sin embargo, tenga presente que lo ideal al crear gráficos como este es irlos construyendo con lo que indican sus estudiantes, aunque es bueno que usted tenga en mente una idea sobre lo que espera incluir ahí.

**Figura 11.** Gráfico de anclaje sesión 1



# Sesión 2

## Aprendizajes esperados

Al final de esta sesión se espera que sus estudiantes puedan:

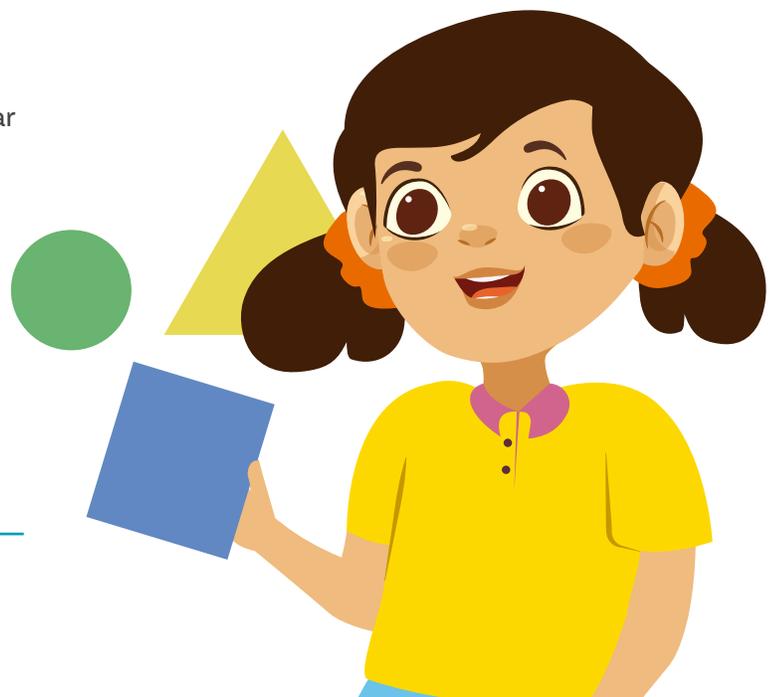
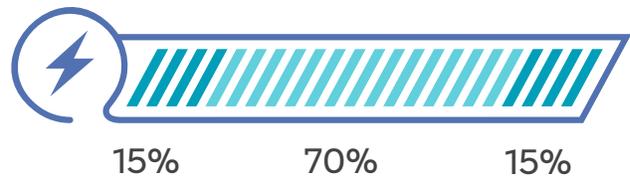


Analizar y corregir errores en un programa que incluya secuencias y bucles.



Seguir instrucciones para desarrollar un código y modificarlo para incorporar bucles sencillos para hacer figuras geométricas.

## Duración sugerida



## Material para la clase

- Anexos 2.1, 2.2, 2.3, 2.4 y 2.5

Anexo

Anexo 1.1

Fichas anotaciones 1	Fichas anotaciones 1
Fichas anotaciones 1	Fichas anotaciones 1
Fichas anotaciones 1	Fichas anotaciones 1
Fichas anotaciones 1	Fichas anotaciones 1

Anexo 1.2

Fichas anotaciones 2	Fichas anotaciones 2
Fichas anotaciones 2	Fichas anotaciones 2
Fichas anotaciones 2	Fichas anotaciones 2
Fichas anotaciones 2	Fichas anotaciones 2

Lo que sabemos,  
lo que debemos saber



Esta sección corresponde al 15% de avance de la sesión

Comience esta sección haciendo un breve resumen de lo aprendido con anterioridad, pídale a sus estudiantes que vuelvan a las fichas de anotaciones 1 y 2 tomadas de los Anexos 1.1 y 1.2.

Pida a sus estudiantes mencionar algunos ejemplos de problemas que han resuelto hasta el momento al programar con Scratch. Mencione, por ejemplo, que en guías anteriores han creado programas que utilizaban notas musicales, operaciones matemáticas y animaciones de la naturaleza. En esta ocasión van a utilizar bucles para crear figuras geométricas.

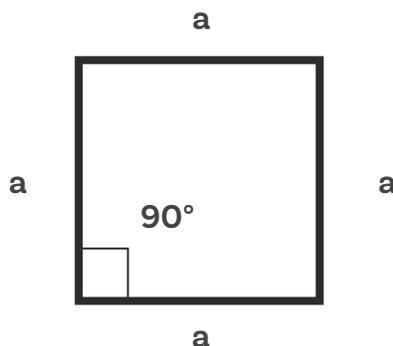
Dibuje un cuadrado en el tablero. Pida a sus estudiantes que escriban el algoritmo para dibujar cuadrados. Si sus estudiantes reconocen que pueden utilizar bucles para simplificar las instrucciones, deténgalos por un momento e invíteles a pensar en las instrucciones sin el uso de bucles. Aproveche este espacio para recordar brevemente que los cuadrados tienen ángulos rectos.

Luego pregunte:



¿Cómo cambiaría el algoritmo para hacer cuadrados grandes o para hacer cuadrados pequeños?  
¿Cómo cambiaría para dibujar rectángulos?

Figura 1. Cuadrado



Dé un tiempo breve a sus estudiantes para pensar en las respuestas y compartir sus ideas con una compañera o compañero. Luego elija a quienes responderán para evitar recibir respuestas en coro. Para seleccionar estudiantes puede utilizar el azar (por ejemplo, con una ruleta de nombres), una dinámica, o seleccionar intencionalmente, idealmente dando la palabra a una niña y un niño cada vez.

Cuando haya escuchado varias respuestas, pida a sus estudiantes que recuerden lo que sus compañeras y compañeros acaban de decir, pues lo usarán en la actividad siguiente. Ahora pase a la actividad conectada.



## Manos a la obra



Esta sección corresponde al 85% de avance de la sesión

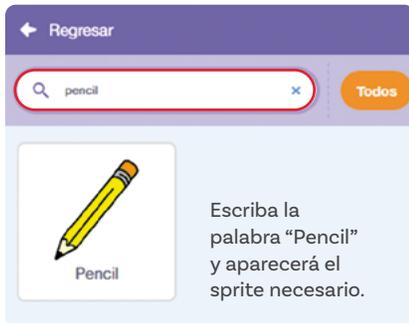
Indique que en la primera actividad van a llevar a *Scratch* su algoritmo para dibujar un cuadrado. Permita que sus estudiantes den ideas acerca de cómo podrán programarlo.

Para este ejercicio, siga estos pasos y pida a sus estudiantes que trabajen, si es posible de forma individual, para replicar lo que usted demuestra en *Scratch*. Si esto no le es posible, implemente la estrategia de programación en pares, asignando roles de piloto(a) y navegante, como se sugirió en la sesión anterior.

Una vez todo el grupo haya logrado ingresar a *Scratch* y crear un nuevo proyecto, explique que van a reemplazar el personaje Gato por un lápiz. Los pasos a seguir son:

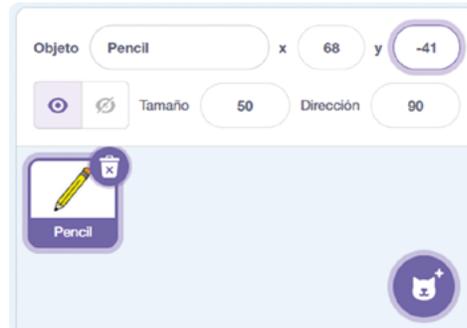
- 1 Borre el personaje de gato presente en el escenario, haciendo clic derecho sobre él y seleccionando “Borrar”, tal como lo realizó en la sesión anterior.
- 2 Ingrese a la biblioteca de personajes y en el cuadro de búsqueda, escriba “Pencil” para encontrar el objeto lápiz. Una vez aparezca, seleccione el objeto “Pencil”. Este le llevará de vuelta al entorno de programación.

**Figura 2.** Seleccionar el objeto lápiz



Luego, ajuste el tamaño del lápiz disminuyéndolo a 50 en la opción “Tamaño”.

**Figura 3.** Pasos para ajustar el tamaño del objeto “Pencil”



**Figura 4.** Bloque girar a la derecha 15 grados



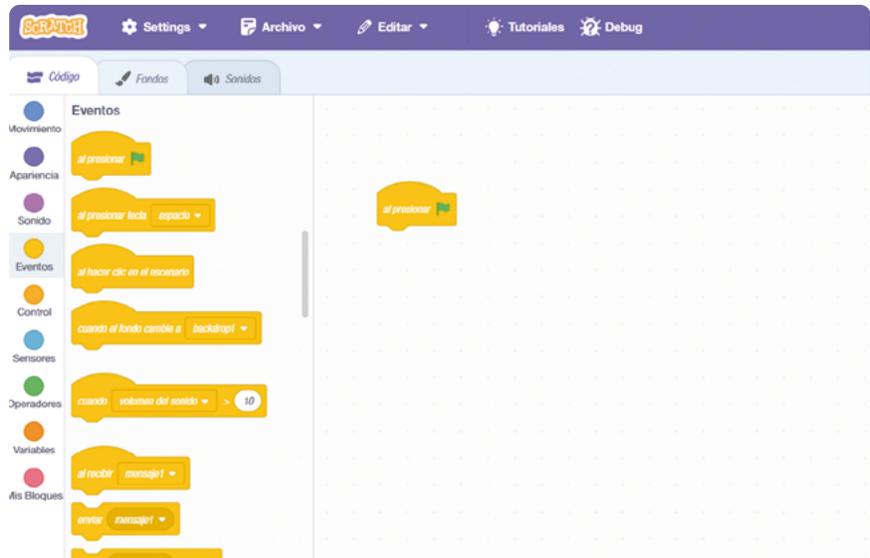
**Figura 5.** Programa para dibujar un cuadrado en Scratch, sin usar bucles



Ahora, proceda a programar el movimiento del objeto "Pencil".

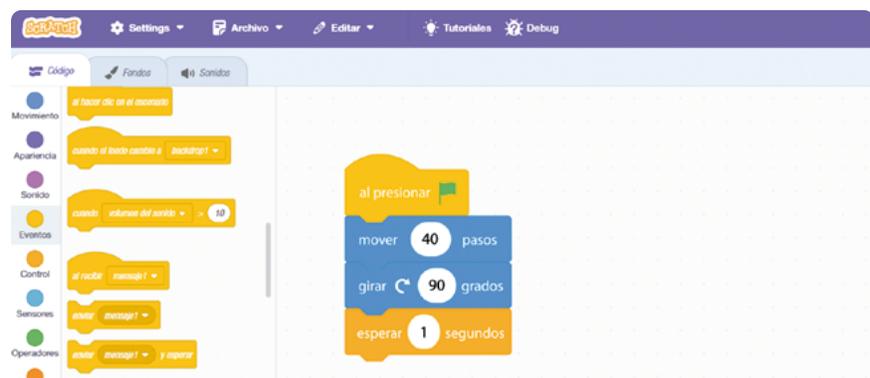
- 1 Haga clic en la pestaña **Código**, categoría **Eventos** y arrastre el bloque **Al hacer clic en la bandera verde** hacia el área de programación, como se ve en la Figura 6.

**Figura 6.** Inicio del código



- 2 Desde la categoría **Movimiento** agregue el bloque **Mover (10) pasos** y cambie el valor 10 por 40. Luego, agregue el bloque **Girar a la derecha (15) grados**, como se ve en la Figura 4 y cambie el valor 15 por 90. Además, desde la categoría de bloques de **Control** agregue el bloque **Esperar (1) segundos**. Su código debería verse como se aprecia en la Figura 7.

**Figura 7.** Avance del programa para dibujar una línea, girar el lápiz 90 grados a la derecha y esperar 1 segundo



- 3 Repita las instrucciones del punto anterior 3 veces más, de modo que el programa quede configurado como se muestra en la *Figura 5*.
- 4 Una vez armado el programa, pida a una o un estudiante que haga clic en la bandera verde mientras los demás observan el comportamiento del lápiz.
- 5 Finalmente, plantee las siguientes preguntas a sus estudiantes para fomentar la reflexión:



¿Qué notan en el comportamiento del lápiz?  
¿Qué hace falta?  
¿Qué ideas tienen para mejorar este programa?

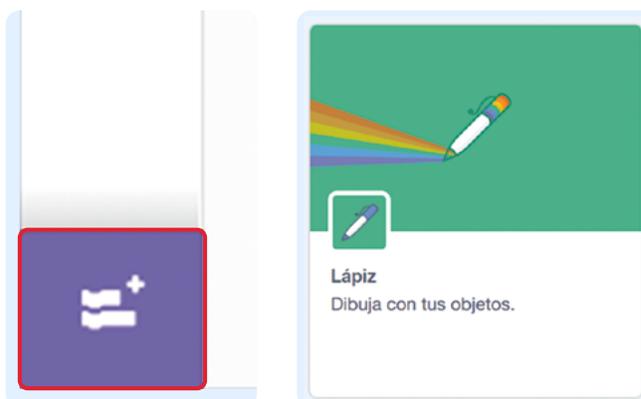
Si dentro de las respuestas no aparece la de “Falta que el lápiz dibuje” usted introducirá esta respuesta:



¿Qué tal si hacemos que el lápiz dibuje?

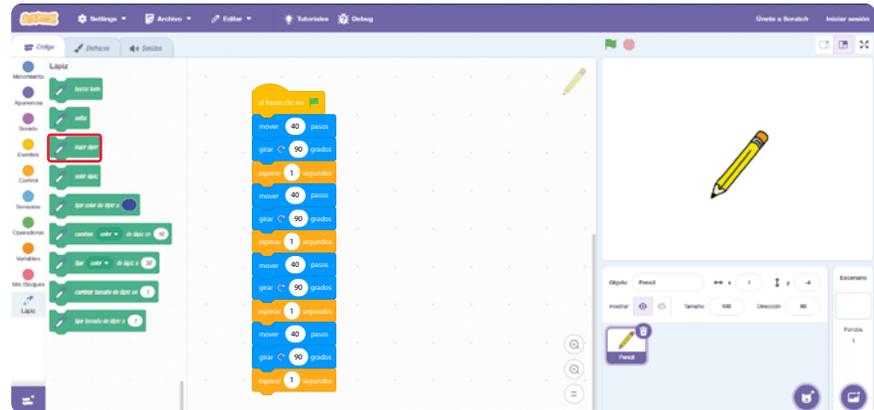
Para continuar, usted debe agregar una extensión llamada “Lápiz”. Para hacerlo, haga clic en el ícono de agregar extensión que encontrará en la parte inferior izquierda del entorno de programación. Esto lo llevará a la ventana de extensiones, donde deberá seleccionar la opción “Lápiz”. Esta acción añadirá los bloques de la extensión lápiz al entorno de programación.

**Figura 8.** Uso de la extensión lápiz



A continuación, seleccione el bloque **Bajar lápiz** para que pueda agregarlo a su programa, tal como se muestra en la Figura 9.

**Figura 9.** Selección del bloque **Bajar lápiz**



Agregue este bloque justo después del bloque **Al hacer clic en bandera verde**, como se observa en la Figura 10.

**Figura 10.** Programa para dibujar un cuadrado, sin bucles en Scratch, utilizando la extensión Lápiz



Pida a una estudiante que haga clic en la bandera verde mientras los demás observan los resultados.

Luego, retome el concepto de bucle y formule la siguiente pregunta a sus estudiantes:



Teniendo en cuenta lo aprendido previamente:  
¿Qué ideas tienen para mejorar el programa?  
¿Dónde se harían estas mejoras? ¿Cómo se harían?

**Figura 11.** Programa en Scratch para dibujar un cuadrado usando bucles



Escuche las respuestas y determine si algunas de ellas están relacionadas con la utilización de bucles. Si se mencionan bucles, guíeles para mejorar el programa utilizando estos conceptos. Si no se hace referencia a ellos, recuérdelos el concepto de bucle y ayúdeles a identificar dónde pueden ser aplicados.

A continuación, pida a sus estudiantes repetir el ejercicio, pero esta vez utilizando bucles, como se muestra en la *Figura 11*.

Para ello, demuestre estos pasos:

- 1 Elimine todos los bloques de programación, excepto los bloques **Al hacer clic en la bandera verde** y **Bajar lápiz**.
- 2 Agregue un bloque **Repetir (10)** y cambie el número 10 por un 4. Recuerde que este bloque lo encontrará en la opción "Control".
- 3 Dentro del bloque **Repetir (4)**, añada un bloque de **Mover (10) pasos** y cambie el 10 por 50, así como un bloque **Girar (15) grados** y cambie el 15 por 90.

De tal manera que el programa se vea como en la *Figura 11*.

Antes de ejecutar el programa, formule las siguientes preguntas a sus estudiantes:

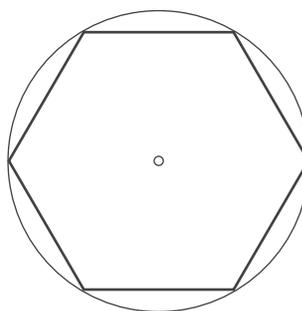


- ¿Qué diferencia notan entre este programa y el anterior?
- ¿Cuál creen que es óptimo y por qué?
- ¿Qué creen que sucederá?
- ¿A qué se debe que eso suceda?

Evalúe las respuestas de sus estudiantes y proporcione retroalimentación, valorando siempre cada respuesta que ofrezcan.

A continuación, guíe a sus estudiantes en el dibujo de un hexágono, que es un polígono de 6 lados. Si le es posible, pídale dibujar con un color diferente una circunferencia alrededor de este, como se observa en la *Figura 12*.

**Figura 12.** Hexágono inscrito en una circunferencia

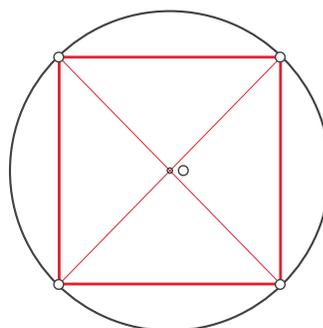


**Nota**

Para dibujar un triángulo se debe repetir el código 3 veces y el ángulo de giro del lápiz debe ser de 120 grados. Para dibujar un pentágono se repiten las instrucciones 5 veces y el ángulo de giro debe ser de 72 grados. Para dibujar un octágono se repiten instrucciones 8 veces y el ángulo de giro debe ser de 45 grados.

Recuérdelos que una circunferencia tiene 360 grados. Recuerde que, para que Scratch hiciera el dibujo del cuadrado se determinó cada giro del lápiz en 90 grados. Dibuje una circunferencia y luego un cuadrado dentro, como se ve en la *Figura 13*. Explique que el giro de 90 grados requerido es el resultado de dividir 360 (los grados de la circunferencia) entre 4 (los lados del cuadrado).

**Figura 13.** Cuadrado inscrito en una circunferencia



Ahora, formule las siguientes preguntas y pida a sus estudiantes pensar sus respuestas antes de levantar la mano para pedir la palabra:



*Si en el caso del cuadrado se repitieron los pasos 4 veces por ser 4 lados, ¿cuántas veces se deben repetir las instrucciones para hacer un hexágono?*

*Si para hacer el cuadrado el ángulo de giro era de 90 grados, ¿De cuántos grados debe ser cada giro para hacer el hexágono?*

*¿Sirve el mismo programa con el que se realizó el cuadrado? ¿Qué habría que cambiarle?*

**Adaptación**

Si es posible, tenga versiones texturizadas del cuadrado, el hexágono y los demás polígonos.

Utilice las respuestas de sus estudiantes para realizar los ajustes necesarios en el programa, de manera que se asemeje al código que aparece en la *Figura 14*.

**Figura 14.** Programa para dibujar un hexágono usando bucles



Pídales a sus estudiantes que repliquen y ejecuten el programa. Luego pídale discutir las preguntas:

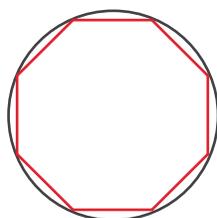
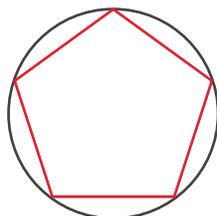
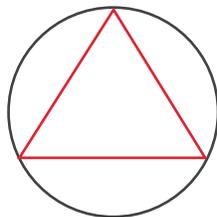


¿Qué cambios realizarían para dibujar un triángulo?  
¿Y si fueran a dibujar un pentágono?  
¿Qué cambios realizarían para dibujar un octágono?

Si tienen dificultades para responder sus preguntas, anímeles a hacer la división respectiva para encontrar los valores exactos de los ángulos de giro que tendrían que hacerse para generar los programas respectivos.

Si cuenta con algo de tiempo, permita que sus estudiantes modifiquen el código para crear las figuras de la *Figura 15*.

**Figura 15.** Polígonos inscritos en circunferencias: triángulo, pentágono y octágono



### Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

Al finalizar la sesión, diga a sus estudiantes que van a hacer un juego para revisar lo que aprendieron a partir de lo realizado en la sesión. En una bolsa o cartuchera, previamente introduzca papeles recortados con las siguientes preguntas u otras semejantes:



*¿Qué diferencias notaron entre el programa que dibujaba un cuadrado usando bloques individuales y el programa que usó el bloque “repetir”?*

*¿Por qué creen que es mejor usar un bloque “repetir” en lugar de escribir los mismos pasos varias veces?*



¿Qué cambios se requerían en el programa para dibujar otras figuras, como triángulos, pentágonos u octágonos?

¿Qué dificultades encontraron al intentar dibujar el hexágono y cómo las abordaron para solucionarlas?

¿Qué estrategias utilizarían en el futuro para solucionar desafíos similares al de dibujar figuras geométricas con Scratch?

¿Cuál creen que es la importancia de comprender el número de repeticiones necesarias y el ángulo de giro al utilizar bucles para dibujar figuras geométricas?

¿Cómo creen que podrían aplicar los conceptos aprendidos en esta sesión en proyectos de programación más complejos o en otros contextos fuera de Scratch?



Circule por el salón dando la oportunidad a diferentes estudiantes de tomar uno de los papeles con preguntas y leerla en voz alta. Invite a esta persona a responder, si siente que puede hacerlo. De lo contrario, dé la oportunidad a otra persona. Trate de alternar participaciones para dar oportunidades equitativas a todo el grupo y que no sean unas pocas personas quienes respondan siempre.

Finalice la sesión completando la cartelera del gráfico de anclaje que empezó la clase anterior. Esto puede hacerlo agregando uno o dos ejemplos de bucles usados para crear polígonos y algunas ideas que surjan a partir de las respuestas de sus estudiantes a las preguntas de cierre anteriores.

# Sesión

# 3

## Aprendizajes esperados

Al final de esta sesión se espera que sus estudiantes puedan:

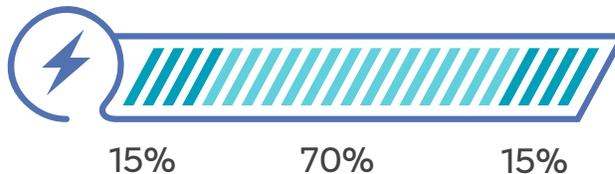


Evaluar y depurar programas que utilizan bucles.



Analizar y aplicar bucles o ciclos para estructurar secuencias repetidas de instrucciones o que deben ejecutarse según una o dos condiciones lógicas.

## Duración sugerida



## Material para la clase

- Material por grupo:** tarjetas previamente recortadas de los Anexos 3.1, 3.2 y 3.3, en cantidad suficiente para dar una tarjeta a cada grupo.



## Anexos

## Anexo 3.1



## Anexo 3.2



## Lo que sabemos,

## lo que debemos saber



Esta sección corresponde al 15% de avance de la sesión

Inicie la clase indicando brevemente a sus estudiantes que deberán utilizar sus habilidades en programación y lo aprendido en sesiones previas para resolver diferentes retos que requerirán el uso de bucles para optimizar programas extensos y crear patrones. Además, retomarán el uso de variables para hacer programas más complejos.

En esta ocasión se les dará una serie de ejercicios donde pondrán en práctica sus conocimientos sobre bucles para resolver una variedad de problemas de programación. Desde dibujar formas geométricas, hasta realizar cálculos matemáticos complejos, aprenderán a utilizar los bucles de manera efectiva para abordar una amplia gama de desafíos.

Manos  
a la obra

Esta sección corresponde al 85% de avance de la sesión

Distribuya a sus estudiantes en grupos de 2 a 3, y entrégueles una de las tarjetas de la ficha de estudio 1 que se encuentran en el Anexo 3.1.

En esta ficha de estudio 1 sus estudiantes encontrarán un programa realizado en Scratch, el cual dibuja un pentágono sin el uso de bucles. Solicite a sus estudiantes que analicen el programa como grupo, identifiquen su función y determinen cómo podrían optimizarlo mediante el uso de bucles.

Tras algunos minutos de discusión grupal, inicie una discusión general. Permita que al menos tres equipos propongan sus mejoras al programa. Proporcione retroalimentación a sus propuestas, destacando siempre los aspectos positivos y fortaleciendo las áreas que puedan presentar debilidades.



Asegúrese de que la organización de los grupos no refuerce brechas existentes. En función de las características de su grupo puede usar estrategias como grupos mixtos, le sugerimos revisar la información en la guía inicial del grado.

Luego de terminar la actividad de la ficha de estudio 1, pase al siguiente ejercicio. Entregue a sus estudiantes una tarjeta recortada con la ficha de estudio 2 que aparece en el Anexo 3.2. Esta ficha muestra requiere que sus estudiantes determinen si el programa propuesto es correcto y que indique, qué debería realizar.

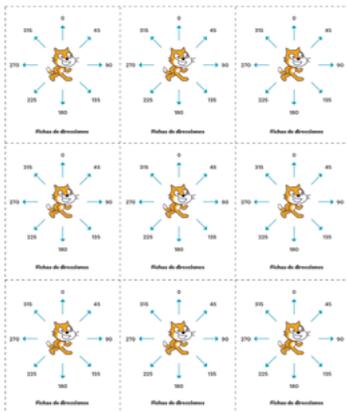
Plantee preguntas como las siguientes y pida a los grupos que las discutan:



- ¿Cuál es la función del programa?
- ¿Se utilizó correctamente el bucle en el programa?
- ¿Qué puede pasar si el objeto toca un borde de la pantalla?
- ¿Cuántas veces se repite el ciclo? ¿Cómo lo saben?

## Anexo

### Anexo 3.3



Para el siguiente ejercicio, entregue a sus estudiantes una tarjeta con la ficha de direcciones que encontrará en el Anexo 3.3 de esta guía. Explique que los objetos en Scratch pueden moverse en direcciones con un rango de 360 grados, pero que hay algunas direcciones comunes que se utilizan frecuentemente para programar el desplazamiento de objetos.

A continuación, muéstreles cómo utilizar estas direcciones a través de un ejercicio en el que un lápiz debe desplazarse por la pantalla dibujando una figura de zigzag. Además, al presionar la tecla de espacio, se borrará lo que se haya dibujado. Reproduzca el código que aparece en la Figura 1.

Explique a sus estudiantes que este programa que permite borrar lo dibujado se ejecuta simultáneamente con el programa que dibuja el zigzag, y que la tecla para borrar puede ser cambiada por cualquier otra que deseen.

Pida a dos estudiantes que pasen al frente para ejecutar el programa mientras los demás observan atentamente. Solicite a las personas que están al frente presionar la tecla de espacio durante la ejecución del programa.

**Figura 1.** Programa para dibujar una línea en zigzag y borrar el dibujo al presionar espacio

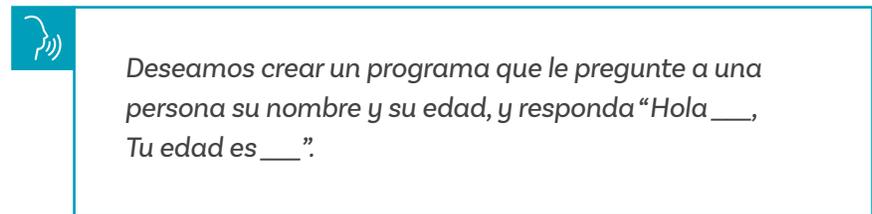


Luego, entregue a cada estudiante una ficha de anotaciones 3, recortada del Anexo 3.4, y pida que escriban una explicación de lo que hace el código.

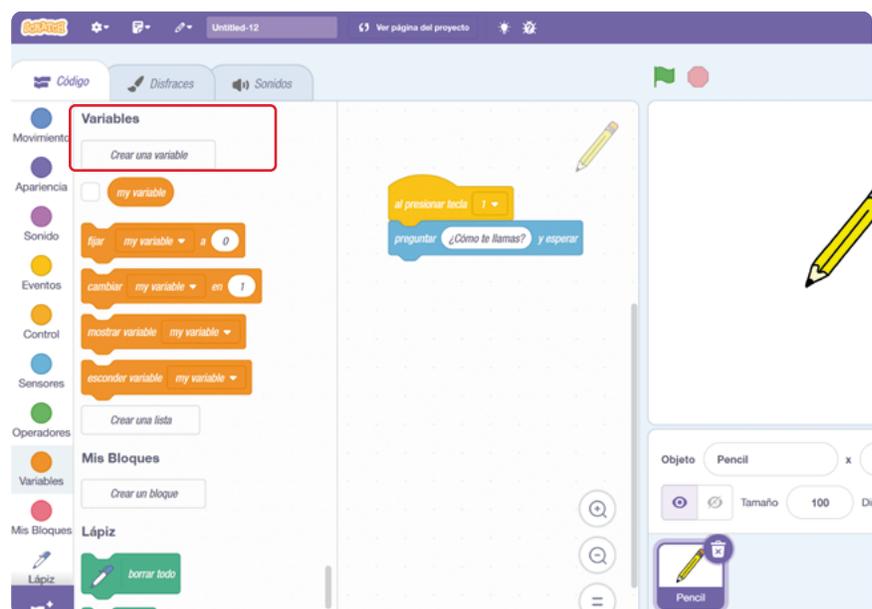
Después de que sus estudiantes diligencien la “Ficha de anotaciones 3”, recuérdelos el concepto de variable que trabajaron en la guía anterior (Guía 3). Pregunte cómo utilizaron las variables y para qué resultaron útiles.

Luego, recuérdelos que, en programación, una variable recibe un nombre y guarda información, como números o palabras. Después, cuando sea necesario utilizar esa información en el programa, se puede preguntar a la variable para obtener una respuesta.

A continuación, recuerde a sus estudiantes cómo usar variables en Scratch a través del siguiente ejemplo:



**Figura 2.** Instrucción para crear variables en Scratch

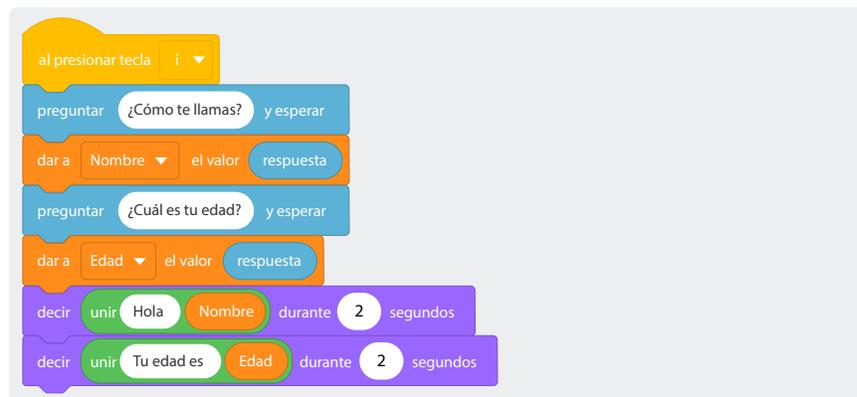


**Figura 3.** Ubicación del bloque **respuesta** en Scratch



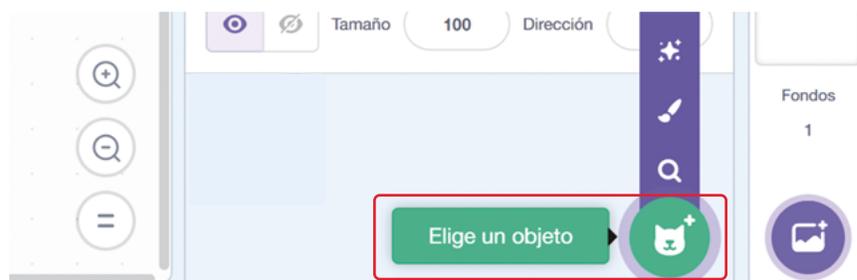
El programa final se muestra en la *Figura 4*. Puede seguir el paso a paso para una explicación detallada.

**Figura 4.** Código en Scratch para preguntar nombre y edad y luego saludar usando esa información. Se hace uso de variables.



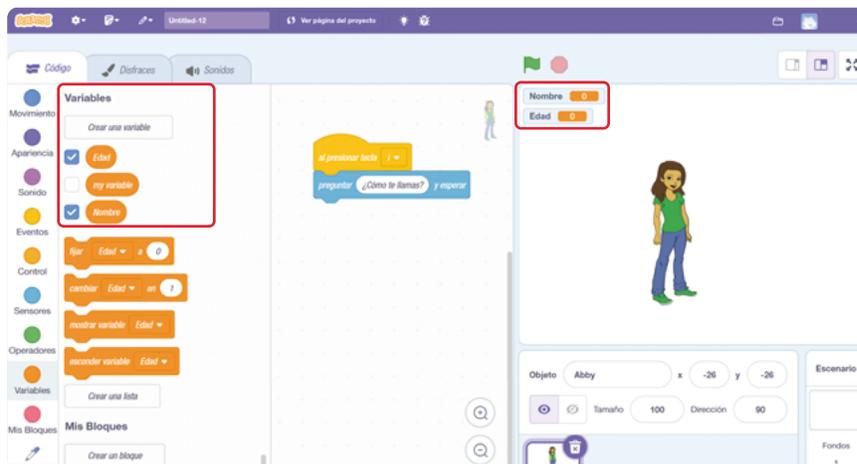
- 1 Cambie al personaje gato por el personaje Abby, dando clic en elegir objeto.

**Figura 5.** Instrucción Elegir un objeto, usada para cambiar el personaje.



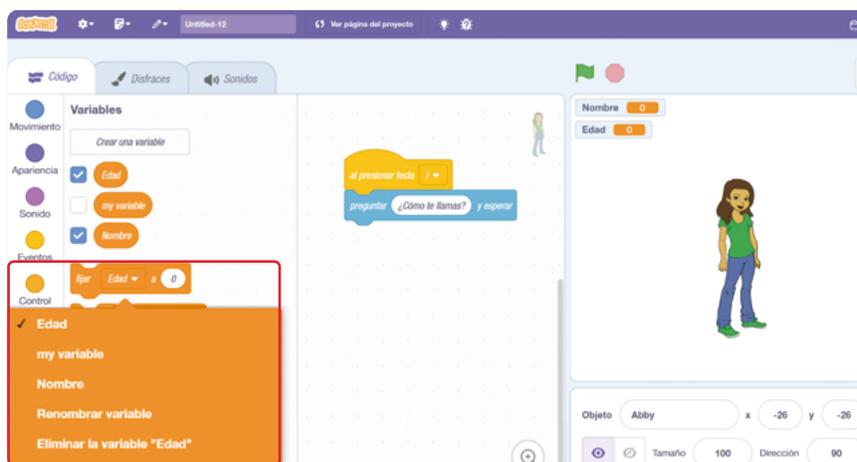
- 2 De la categoría **Eventos**, elija el bloque **Al presionar tecla (Espacio)** y agréguelo al entorno de programación. Luego, cambie "Espacio" por la letra "I".
- 3 En la categoría **Sensores** elija y agregue el bloque **Preguntar () y esperar**. Dentro de los paréntesis escriba "¿Cómo te llamas?".
- 4 Agregue dos variables haciendo clic en **Agregar Variable** de la categoría **Variables**, tal como lo muestra la *Figura 2*. La primera variable se llamará "nombre" y la segunda, "edad". Cuando lo haya hecho, la interfaz se verá como la *Figura 6*.

**Figura 6.** Variables nombre y edad creadas en programa, aunque sin vincular al código todavía. Se observa que aparecen también en la escena



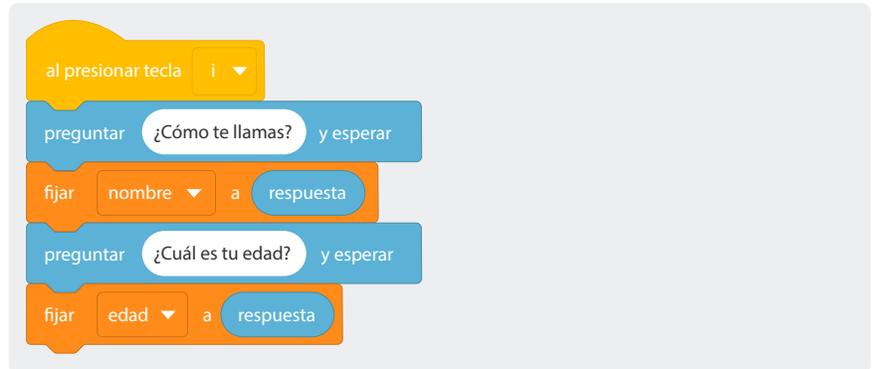
- 5 De la categoría **Variables**, agregue el bloque **Dar a (A) el valor (B)**. Tenga en cuenta que, dependiendo de la versión de Scratch que esté usando, este bloque podría llamarse **Fijar (mi variable) a (O)**, como se observa en la Figura 7.

**Figura 7.** Ubicación del bloque **Fijar (mi variable) a (O)**, también llamado **Dar a (A) el valor (B)**



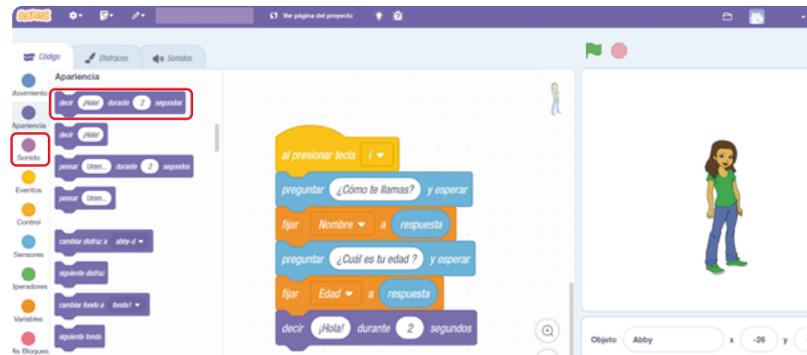
- 6 Reemplace el nombre de la variable que aparece por defecto (“A” o “mi variable”) por la variable “Nombre” y el valor a asignar (“B” o “0”) por el bloque **Respuesta** que encontrará en la categoría **Sensores** como muestra la Figura 3. Repita estos pasos para la variable “Edad”. Su código en este punto debería verse como en la Figura 8.

**Figura 8.** Programa que pregunta nombre y edad y almacena las respuestas en las respectivas variables.

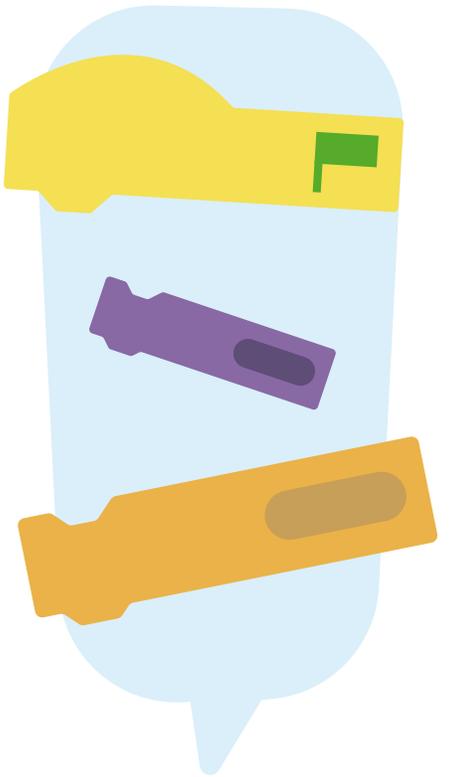
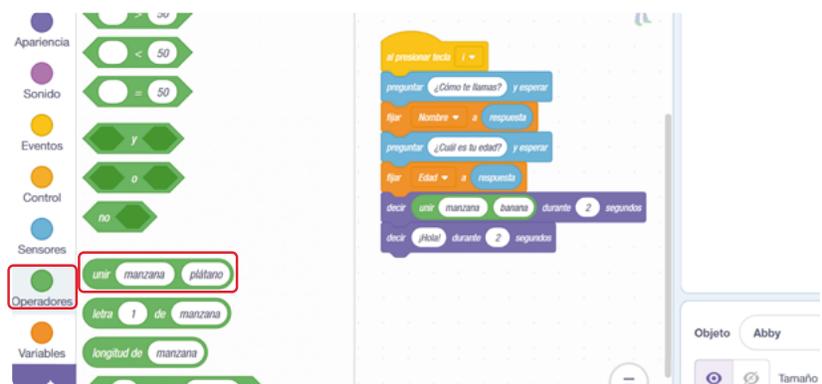


- 7 Agregue dos bloques **Decir () durante (2) segundos**, que se encuentran en la categoría Apariencia, como se ve en la Figura 9. A cada bloque, añada el bloque **Unir (Manzana) (Plátano)**, que encontrará en la categoría Operadores, como se ve en la Figura 10.

**Figura 9.** Ubicación del bloque **Decir () durante (2) segundos**

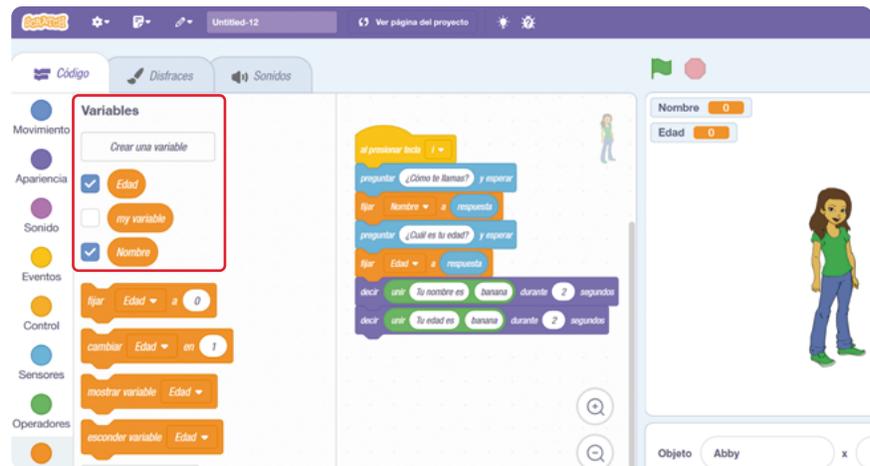


**Figura 10.** Ubicación del bloque **Unir (Manzana) (Plátano)**



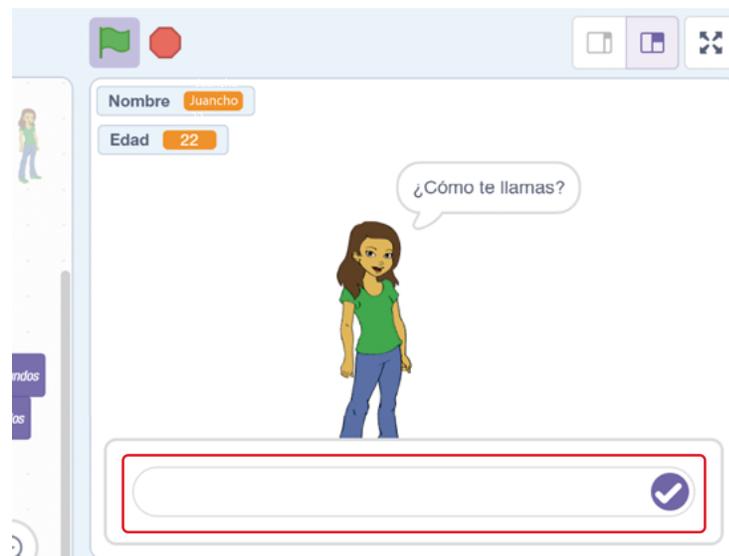
- En el primer bloque, reemplace la palabra “Manzana” por “Tu nombre es:” y en el segundo bloque, reemplace la palabra “Manzana” por “Tu edad es:”.
- En ambos bloques, reemplace “Plátano” por la variable correspondiente: “nombre” en el primer bloque y “edad” en el segundo, como se ve en la *Figura 11*.

**Figura 11.** Ubicación de la variable edad



Pida a una estudiante que ejecute el programa y que llene los datos que le piden en la caja correspondiente para ingreso de información, como muestra la *Figura 12*.

**Figura 12.** Localización de la caja de ingreso de información



## Anexo

## Anexo 3.5

	<p><b>Hoja de trabajo 1</b></p> <p>Analice el programa entregado y deduzca cuál es su función. Escriban sus notas al respecto.</p> <p>-----</p> <p>Ahora, repliquen el programa en Scratch y añadan su funcionamiento.</p> <p>¿Coincide con lo que hablan pensando? (¿Y qué es diferente?)</p> <p>Usando como modelo este programa, se debe crear un programa que permita determinar el promedio de los calidos de un grupo de estudiantes de un curso. Recuerden preguntar cualquier duda antes de ir al grupo.</p>
	<p><b>Hoja de trabajo 2</b></p> <p>Analice el programa entregado y deduzca cuál es su función. Escriban sus notas al respecto.</p> <p>-----</p> <p>Ahora, repliquen el programa en Scratch y añadan su funcionamiento.</p> <p>¿Coincide con lo que hablan pensando? (¿Y qué es diferente?)</p> <p>Usando como modelo este programa, se debe crear un programa que permita determinar el promedio de los calidos de un grupo de estudiantes de un curso. Recuerden preguntar cualquier duda antes de ir al grupo.</p>
	<p><b>Hoja de trabajo 3</b></p> <p>Analice el programa entregado y deduzca cuál es su función. Escriban sus notas al respecto.</p> <p>-----</p> <p>Ahora, repliquen el programa en Scratch y añadan su funcionamiento.</p> <p>¿Coincide con lo que hablan pensando? (¿Y qué es diferente?)</p> <p>Usando como modelo este programa, se debe crear un programa que permita determinar el promedio de los calidos de un grupo de estudiantes de un curso. Recuerden preguntar cualquier duda antes de ir al grupo.</p>

Pregúnteles a sus estudiantes lo siguiente:



- ¿Qué tarea o tareas realiza el programa?
- ¿Cuál es papel que juegan las variables “Nombre” y “Edad”?
- ¿Qué papel juega el bloque “Dar a (Variable) el valor (Respuesta)”?
- ¿Cuál es la función del bloque “Unir”?
- ¿Qué ejemplos podríamos usar para incluir un bucle?

Ahora, recuérdelos que las variables pueden almacenar tanto palabras como números y con estos últimos se pueden realizar operaciones matemáticas como sumas, restas, multiplicaciones y divisiones.

Entregue a cada grupo una tarjeta de la guía de trabajo 1 que aparece en el Anexo 3.5 y permita que la trabajen de manera autónoma. Tenga en cuenta que esta guía incluye tanto un ejercicio de lectura de código como de creación de un nuevo programa que hace cálculos con variables. Monitoree los grupos y aclare dudas si se requiere, permitiendo, no obstante, que sus estudiantes sean quienes crean el programa.

Reserve unos minutos para hacer el cierre de la sesión.

## Glosario



**Variable:** elemento que almacena un valor que puede cambiar durante la ejecución de un programa o experimento.

## Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

Al finalizar la sesión, invite a sus estudiantes a reflexionar sobre lo aprendido. Puede hacerlo utilizando una “Caja de sorpresas” donde puede poner papeles con preguntas como:



*¿Es más fácil leer un programa que usa bucles?  
¿Qué desafíos encontraron al analizar programas para identificar dónde podrían usarse bucles?  
¿Qué importancia tiene comprender el papel de las variables en la programación?*

Además, puede poner en la caja de sorpresas otras actividades como: hacer un resumen de la clase en pocas palabras o hacer en 5 minutos un dibujo que represente los aprendizajes. Cada grupo debe sacar un papel que describa la pregunta o actividad, y en caso de que sea una pregunta, se puede asignar un tiempo para que el grupo analice lo que se pregunta y piense en la respuesta de manera colectiva.

Discuta con sus estudiantes cómo los bucles permiten automatizar tareas repetitivas, facilitando la creación de programas más eficientes y claros. Además, explique que, al evitar la repetición manual de instrucciones, el código se vuelve más sencillo y organizado. Relacione esto con ejemplos cotidianos como *poner la mesa para cuatro personas*, donde se repiten las mismas acciones varias veces: colocar un plato, un vaso, y cubiertos para cada persona.



# Sesión 4

## Aprendizajes esperados

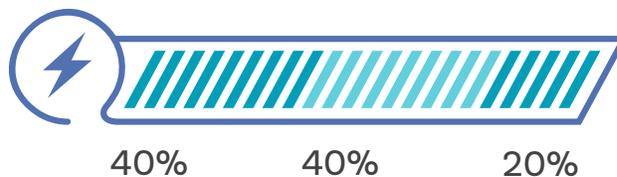


Crear programas sencillos que usan bucles, en un lenguaje de bloques.



Crear, analizar y optimizar un programa en un lenguaje de bloques utilizando secuencias de instrucciones.

## Duración sugerida



Al final de esta sesión se espera que sus estudiantes puedan:

## Material para la clase

- **Material por grupo:** Tarjetas ficha de anotaciones 4.
- Anexo 4.1 a recortar uno por grupo.



**Nota**

En grados superiores se fortalecerá el concepto de condicional, esta es una introducción preliminar para explorar cómo se controlan los programas.

**Lo que sabemos,  
lo que debemos saber**

Esta sección corresponde al 40% de avance de la sesión

Inicie esta nueva sesión repasando con sus estudiantes lo aprendido hasta ahora. Pídales que, en sus propias palabras, expliquen qué es un bucle y cómo lo han utilizado en *Scratch*.

Recuerde a sus estudiantes que los bucles son una serie de acciones que se repiten a lo largo del tiempo. Desde una perspectiva más técnica, un bucle es un conjunto de instrucciones que se ejecutan repetidamente mientras se cumple una determinada condición.

Introduzca el concepto de **condicional** como una estructura de programación que permite que una parte del código se ejecute solo si se cumple o no una cierta condición. Ambos conceptos –bucles y condicionales– están estrechamente relacionados y muchas veces se utilizan juntos para optimizar el comportamiento de los programas.

Para contextualizar a sus estudiantes, plantee el siguiente ejemplo:

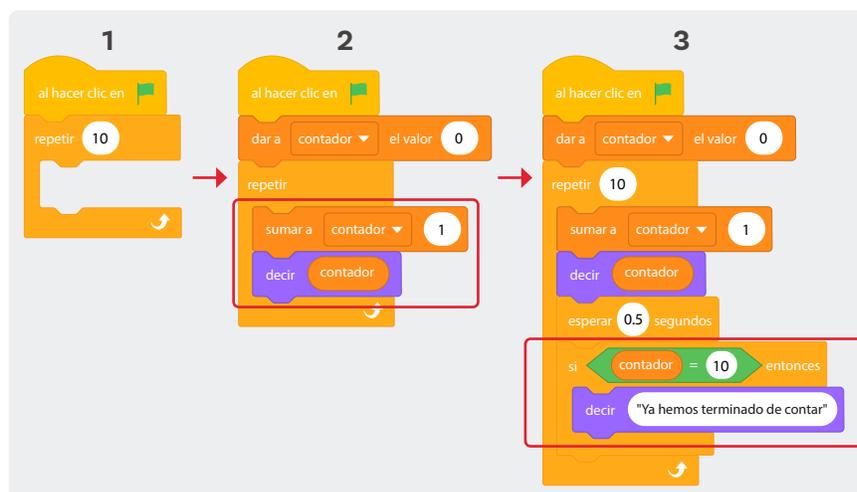


*Supongamos que queremos que el personaje (el gato) cuente del 1 al 10, y cuando llegue al final del conteo, muestre un mensaje diciendo “Ya hemos terminado de contar”.*

A continuación, muéstrelas cómo el programa evoluciona con la inclusión del condicional, subrayando cómo esta mejora el control sobre la ejecución del código.



Figura 1. Código con condicional



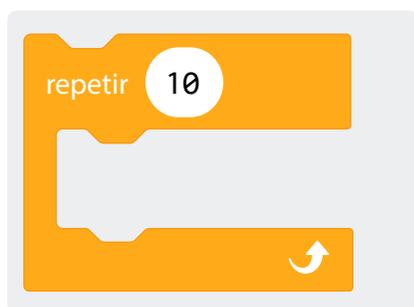
Explique que en el primer ejemplo utilizarán el bloque **repetir x** y luego conocerán un nuevo tipo de bucle. Este bloque permite que cualquier acción dentro de él se ejecute un número determinado de veces (por defecto, 10). Esto definirá cómo se realizará el conteo, y lo único que necesitamos es que el personaje muestre este conteo en pantalla mediante un bocadillo de diálogo.

Para hacerlo, utilizaremos el bloque **decir**, que despliega un mensaje con el texto que escribamos en él.

Es importante que sus estudiantes comprendan que el conteo que queremos mostrar se realiza de manera interna por *Scratch* durante cada iteración del bucle. Sin embargo, para que podamos visualizar el valor del conteo, necesitamos una variable que almacene este número.

Por eso, crearemos una variable llamada **contador**, la cual empezará en 0 y se incrementará en cada repetición del bucle. Esto puede lograrse utilizando el bloque **sumar a VARIABLE**, que permite incrementar o decrementar el valor de una variable en la cantidad que elijamos. Como deseamos contar de uno en uno, configuramos el valor en 1.

Con este código, el personaje será capaz de contar del 1 al 10 sin problemas. Sin embargo, nuestro objetivo es que, al terminar de contar, el personaje muestre un mensaje indicando que ha finalizado, y es aquí donde entra el uso de los condicionales.

**Figura 2.** Bloque repetir**Figura 3.** Bloque por siempre**Figura 4.** Repetir hasta que

Para ello, utilizaremos el bloque **Si... entonces**, que es uno de los condicionales más simples. Sabemos que el conteo termina cuando el contador alcanza el valor de 10, por lo que usaremos este condicional para comprobar si el programa ha finalizado el conteo. Si es así, el personaje mostrará un mensaje en pantalla indicando que ha completado la tarea.

Para profundizar en el funcionamiento de los bucles y los condicionales, es importante analizar cada uno en detalle. A continuación, presénteles los distintos tipos de bucles que Scratch ofrece por defecto y pida que sus estudiantes expliquen las diferencias entre ellos.

Las siguientes definiciones pueden servirle de guía:

**Repetir:** este bucle ejecuta lo que se coloque en su interior un número finito de veces. El número máximo de iteraciones puede establecerse de forma estática (como se muestra en la *Figura 2*), o puede modificarse durante la ejecución del programa en respuesta a eventos externos.

**Por siempre:** similar al bucle **Repetir**, pero con una diferencia clave: en este caso, las instrucciones dentro del bucle se ejecutan indefinidamente, sin un número fijo de repeticiones, hasta que el programa se detenga manualmente.

**Repetir hasta que:** es una combinación de los dos anteriores. Este bucle repetirá las acciones en su interior indefinidamente, pero detendrá la ejecución una vez que se cumpla una condición específica.

Ahora, invite a sus estudiantes a integrar varios de estos conceptos previos mediante la creación de un programa en Scratch. El objetivo del programa será que el personaje siga el puntero del ratón a través de la pantalla, lo que permitirá que pongan en práctica los bucles y condicionales aprendidos.

**Figura 5.** Programa que usa el bucle **Repetir hasta que** y un condicional de salida, y hace que el personaje esté animado y persiga el puntero del ratón



## Glosario

- 
**Condicional:** estructura en programación que permite ejecutar ciertas acciones solo si se cumple una condición específica.
- 
**Contador:** variable que se usa para llevar un registro de cuántas veces ocurre una acción o se repite un proceso, aumentando o disminuyendo su valor según lo indicado en el programa.

## Manos a la obra



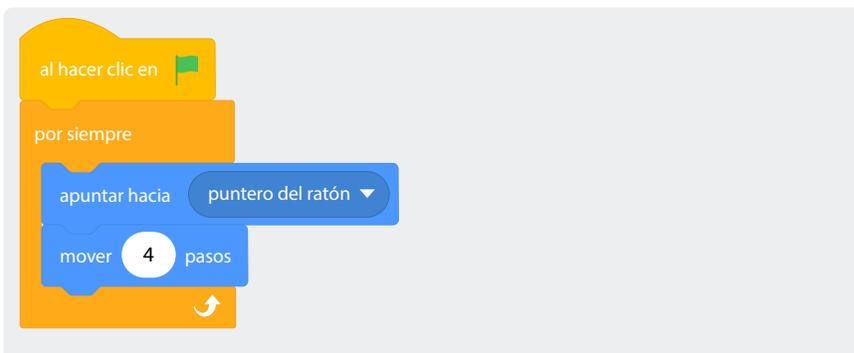
Esta sección corresponde al 80% de avance de la sesión

Pida a sus estudiantes que ingresen a *Scratch* y sigan sus indicaciones para realizar el siguiente ejercicio de manera individual.

- 1 Cambien el objeto que aparece por defecto, por un globo en la galería.
- 2 Agreguen el bloque **Al hacer clic en bandera verde**.
- 3 Debajo, agreguen un bloque **Por siempre**, luego dentro de este bloque, inserten un bloque **Apuntar hacia (Puntero del ratón)**.
- 4 Debajo de este, coloquen un bloque **Mover (10) pasos** cambiando el 10 por un 4.

El programa debe quedar como se muestra en la *Figura 6*.

**Figura 6.** Programa que hace que el personaje persiga el puntero del ratón de forma continua. Se usa el bloque Por siempre



Pídale a una de sus estudiantes que ejecute el programa y realice las siguientes preguntas al público:



- ¿Cuál es la funcionalidad del bloque “Por siempre”?
- ¿Cuántas veces crees que se repetirán las instrucciones que se encuentran dentro de ese bloque?
- ¿En qué momento o lugar de la vida sucede esto?
- ¿Qué ejemplos de procesos infinitos mencionamos en clases anteriores?

Modifique el código para recrear el programa de la Figura 7. Pregunte a sus estudiantes qué creen que pasará al ejecutarse.



**Figura 7.** Programa que hace uso del bloque Por siempre y hace que el personaje, animado con cambio de disfraces, persiga el puntero del ratón



Ejecute el programa y permita que comparen con sus predicciones.

Luego, proceda a presentar los bloques para validar condiciones. Indíqueles que, para evaluar las condiciones, se utilizan unos bloques especiales que encuentra en la categoría **Operadores**. Sus funciones básicas se describen a continuación:



El bloque comprueba si el primer valor es menor que el segundo. Si es menor, el bloque devuelve verdadero (true); si no, devuelve falso (false). Este bloque también funciona con letras, además de con números. En *Scratch*, las letras de la parte superior del alfabeto (por ejemplo, a, b, c) valen menos que las del final (por ejemplo, x, y, z).



El bloque comprueba si el primer valor es mayor que el otro. Si el segundo valor es menor, el bloque devuelve verdadero (true); si no, devuelve falso (false).



El bloque comprueba si el primer valor es igual al otro. Si los valores son iguales, el bloque devuelve verdadero (true); si no, falso (false). Este bloque no distingue entre mayúsculas y minúsculas.

Ahora muéstreles cómo se usan estos condicionales a través de la modificación del programa que realizó anteriormente donde animaba el globo.

## Anexo

## Anexo 4.1

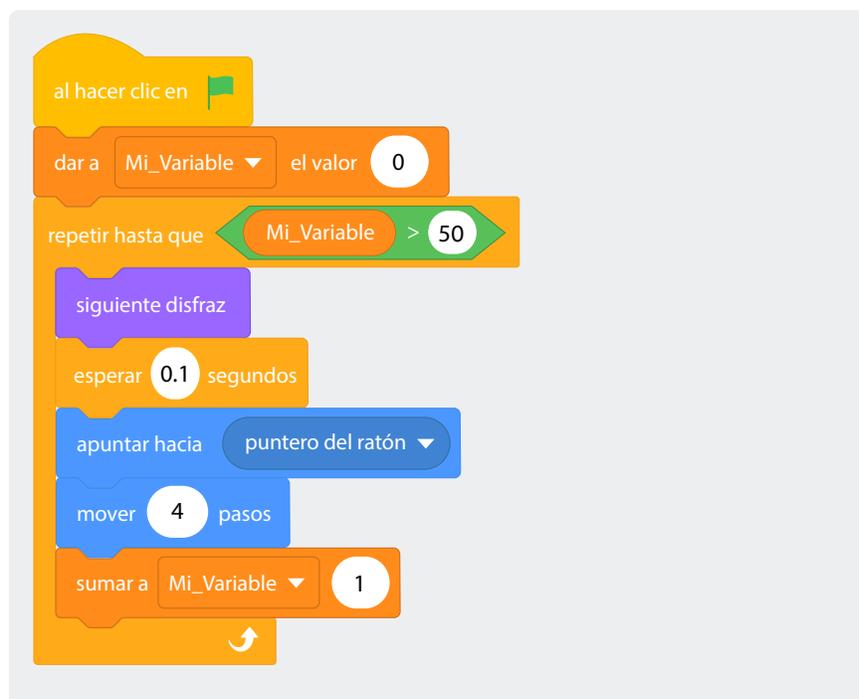


El programa debe quedar tal como se muestra en la *Figura 3*, note que el bloque **Por siempre** fue cambiado por un bloque **Repetir hasta que (Condicional)** y se agregó una variable llamada “Mi\_Variable”, que es la que será evaluada en el condicional.

Después, muéstreles a sus estudiantes lo que sucede al colocar el operador condicional “**Mayor que**” en el lugar correspondiente del bucle, tal como lo muestra la *Figura 8*, recuerde agregar la variable “Mi\_Variable” al operador.

Entregue la ficha de anotaciones 4 del Anexo 4.1 a sus estudiantes y pídeles que escriban lo que sucede al ejecutar el programa, pídeles que estén pendientes del valor de la variable “Mi\_Variable”.

**Figura 8.** Programa que usa el bucle **Repetir hasta que**, y el condicional **mayor que (>)**



Luego, cambie el bloque del operador “**Mayor que**” por el bloque “**Igual que**” como se muestra en la *Figura 9*. Pídeles a sus estudiantes que realicen las observaciones de los valores de la variable “Mi\_Variable” y las plasmen una tarjeta recortada de la ficha de anotaciones 4, Anexo 4.1.

**Figura 9.** Programa que usa el bucle **Repetir hasta que**, y el condicional **igual a(=)**



Por último, cambie nuevamente el bloque operador “**Igual que**” por el bloque “**Menor que**”, ejecute el programa y pídale a sus estudiantes que anoten sus observaciones.

Realice las siguientes preguntas:



- ¿Qué sucede con el programa?
- ¿Por qué no se detiene?
- ¿Qué cambiarían para que este se detenga?

Ahora, como ejercicio, pídale que generen un programa que pregunte por una letra, la almacene en una variable llamada “Letra”, este programa se ejecutará por siempre y solo se detendrá cuando la persona escriba la letra “P”.

## Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

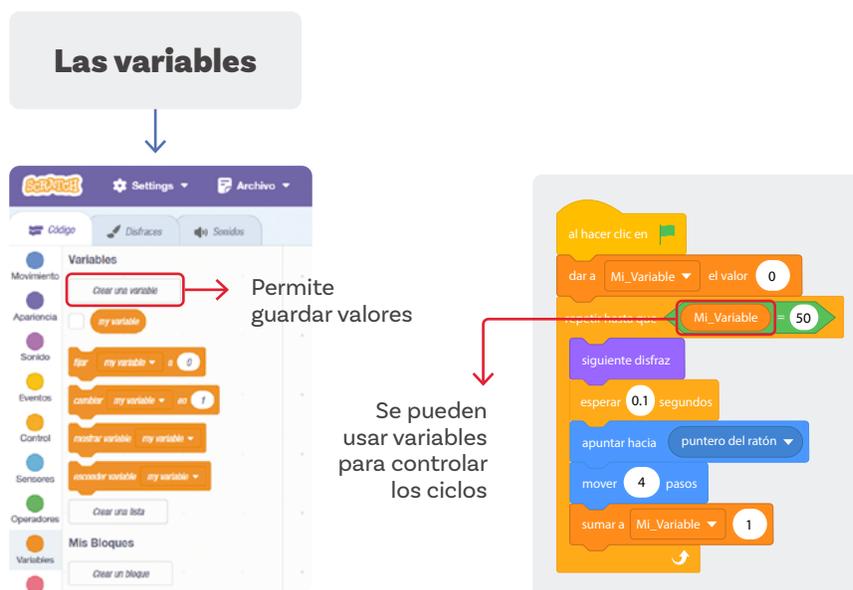
Tras realizar los programas anteriores, pida a sus estudiantes que en parejas completen las siguientes frases:

- Los bucles y los condicionales ayudan a hacer más fáciles de escribir y modificar los programas porque \_\_\_\_\_
- Al crear juegos o programas que requieren que se repitan algunas acciones varias veces es importante usar \_\_\_\_\_ porque \_\_\_\_\_
- La diferencia entre un bucle **repetir ( )** y un **bucle repetir hasta que (condición)** es que \_\_\_\_\_

Escuche las frases completadas por varias parejas y use esa información para determinar si debe hacer alguna aclaración o reforzar algo antes de finalizar.

Concluya la sesión completando el gráfico de anclaje con la información que le provean sus estudiantes.

**Figura 10.** Grafico de anclaje sesión 4



# Sesión 5

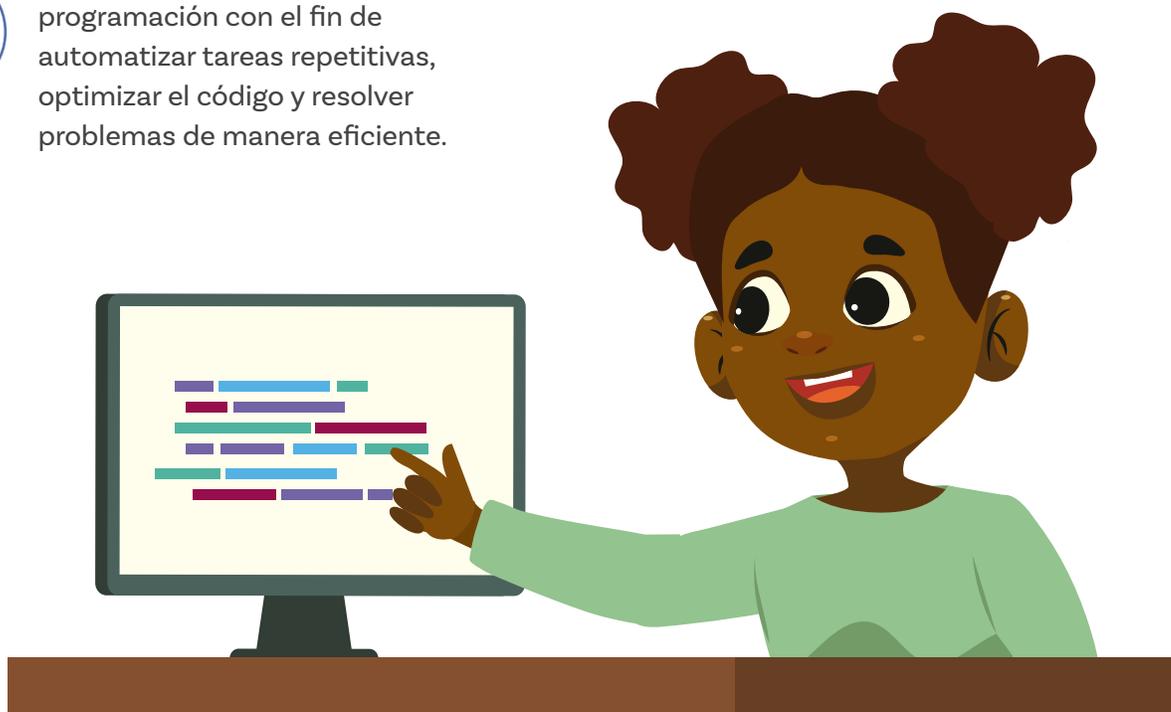
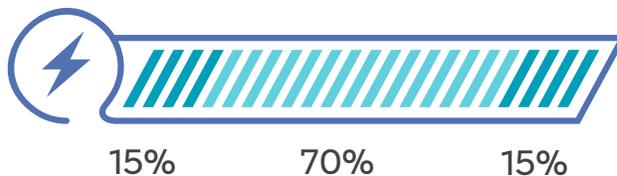
## Aprendizajes esperados

Al final de esta sesión se espera que sus estudiantes puedan:



Usar los bucles en programación con el fin de automatizar tareas repetitivas, optimizar el código y resolver problemas de manera eficiente.

## Duración sugerida





En muchas aulas, cuando se hace una pregunta al grupo, pasan pocos segundos antes de que sus estudiantes que saben más o se sienten más confiados respondan y esto da poca oportunidad al resto del grupo. Con frecuencia, quienes responden primero son niños porque se sienten más cómodos en ambientes de competencia, de modo que pedir a toda la clase que primero piense en silencio, ayuda a que realmente toda la clase se involucre en el análisis de la pregunta realizada.

## Lo que sabemos, lo que debemos saber



Esta sección corresponde al 15% de avance de la sesión

Inicie la sesión recordando lo que han desarrollado en sesiones anteriores. Dé un espacio a sus estudiantes para que hagan preguntas sobre lo que han hecho y para resolver dudas que han surgido en el proceso.

Invite a sus estudiantes a responder mentalmente las siguientes preguntas:



- ¿Qué me llama la atención de la programación?
- ¿Qué se me ha dificultado?
- ¿Qué he aprendido?
- ¿Qué necesito comprender mejor?

Proponga que escriban sus dudas en papeles pequeños que usted podrá responder de manera anónima, explicando que muchos compañeros y compañeras pueden tener las mismas preguntas, aunque no se atrevan a expresarlas.

Luego, retome los conceptos de bucles y condicionales, subrayando que, en *Scratch*, los bucles son esenciales para crear animaciones, juegos y aplicaciones interactivas. Estos permiten, por ejemplo, que un personaje se mueva repetidamente en una dirección, que un objeto cambie de color en intervalos regulares, o que un enemigo siga un comportamiento predecible en un juego. Sin bucles, tendríamos que escribir mucho más código, lo que complicaría el programa.

A su vez, los condicionales permiten que los programas tomen decisiones, como elegir una opción en un menú o reaccionar a una acción. Ambos conceptos son fundamentales y están presentes en programas, videojuegos y dispositivos electrónicos como videoconsolas, teléfonos y televisores, ya que permiten que los sistemas respondan y se adapten a eventos como pulsar una tecla o mover un(a) personaje.

**Figura 1.** Posible personaje y fondo para ilustrar ejemplo de narrativa



**Figura 2.** Código 2



Para esta parte de la sesión, ya usted ha mostrado todas las opciones necesarias para que sus estudiantes utilicen los bloques de bucles acompañados o no de condicionales. Por lo tanto, esta sección es un refuerzo de los conocimientos.

## Manos a la obra



Esta sección corresponde al 67% de avance de la sesión

En grupos de tres, presente a sus estudiantes los códigos en las Figuras 1, 2 y 3. Pídeles que expliquen lo que hará cada uno. En sus cuadernos deben escribir lo que hacen y resaltar las diferencias entre los tipos de ciclos.

Después de verificar las diferencias entre los tres tipos de bucles que se utilizan, destaque también el uso de variables, en caso de que no haya sido mencionado por sus estudiantes.

Proponga a sus estudiantes que diseñen una animación creativa que cumpla con los siguientes requerimientos:

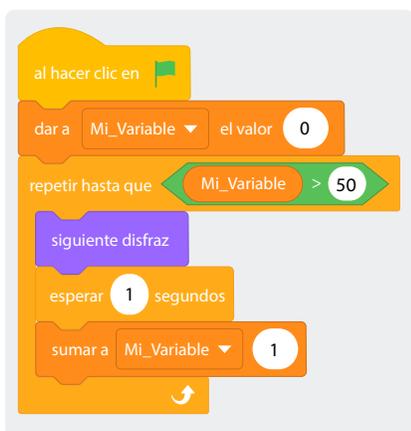
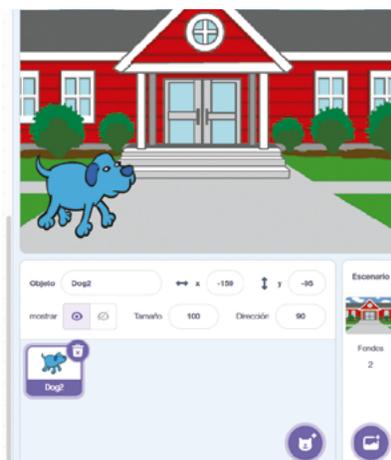
- 1 Debe incluir al menos **un bucle** para realizar una acción repetitiva.
- 2 Debe usar una **variable** para almacenar información, como un contador o un valor relacionado con la animación.

Para apoyar la creación de esta animación, invíteles a hacer una lluvia de ideas sobre los tipos de animaciones que podrían crearse cumpliendo con los requerimientos previamente indicados.

Mencione, por ejemplo:

- Un(a) personaje que cuenta de 1 en 1 utilizando un bucle y que solo se detiene cuando se le da una palabra secreta.
- Un(a) personaje que se mueve alrededor de la pantalla, mientras un contador registra y muestra en pantalla cuántas vueltas completa.

Escriba estas ideas en el tablero.

**Figura 3.** Código 3**Figura 4.** Posible personaje y fondo para ilustrar ejemplo de narrativa

Luego, pida a sus estudiantes que elijan alguna de las ideas en el tablero y que piensen en la historia detrás de esa animación. Dé un ejemplo. Podría mencionar que la persona que se mueve alrededor de la pantalla es una deportista que está practicando para una maratón y que, por eso, se cuentan las vueltas que va completando.

Permita unos minutos para que sus estudiantes propongan otras narrativas o historias que podrían conectar con algunas de las ideas en el tablero.

Cuando haya al menos unas 3 o 4 ideas de posibles animaciones, con narrativas conectadas, invite a sus estudiantes a dedicar unos minutos para elegir una de estas y continuar la planeación de su animación. Pídales elegir los personajes y el fondo que utilizarán según la narrativa que hayan decidido usar para crear su animación.

Presente una propuesta. Por ejemplo, usted podría mencionar que pensaba usar el fondo del colegio (School) y el personaje perro (Dog2), para animar la historia de Rufo, un perrito guardián que vigila la entrada de la escuela varias horas al día. Indique que, en su propuesta, Rufo preguntará cuántas horas tendrá que vigilar y después de que se le indique un número, Rufo se moverá de un lado a otro de la pantalla hasta que se completen las horas que va a trabajar ese día.

Luego escuche a algunos grupos reportarle sus decisiones sobre las(los) personajes y fondo que han pensado utilizar. Pregunte también cómo usarán los bucles y la variable.



¿Qué personajes utilizarán?  
¿En qué fondo ocurrirá la animación?

Indique a sus estudiantes que son libres de elegir la narrativa que habían pensado originalmente u otra que haya sido planteada por alguien más, si esa les pareció más interesante.

Anime a los grupos a empezar sus animaciones mientras usted circula, escuchando las ideas de quienes no lograron compartir previamente.

Ofrezca ayuda si se requiere y monitoree el trabajo, asegurándose de que sus estudiantes aplican los conceptos aprendidos. Cuando se acerque a los grupos, hágales preguntas para que le expliquen cómo utilizan los bucles, variables y contadores en su animación.

## Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

Invite a sus estudiantes a pensar en lo que aplicaron y reforzaron con la experiencia de diseño de la animación.

Puede hacerlo utilizando un ejercicio que requiere respuesta física. Consiste en que usted dice una frase, quienes creen que es correcta deben ponerse de pie, quienes creen que es falsa deben quedarse sentados. Luego, diga frases como las siguientes:

- Un bucle **Repetir ()** en Scratch permite que una acción se ejecute solo una vez. (F)
- Las variables pueden almacenar tanto números como letras o palabras. (V)
- Las instrucciones o bloques que están dentro de un bucle **por siempre** se ejecutan hasta que se finalice el programa. (V)
- El bloque **repetir hasta que** requiere una condición que detenga las repeticiones. (V)

- Si hay bucles, los condicionales son opcionales en los programas. (F)

Aproveche las respuestas de sus estudiantes para identificar los aspectos que aún se les dificultan. Destaque, asimismo, que las **variables**, los **bucles** y los **condicionales** son elementos fundamentales en la programación, ya que trabajan juntos para crear programas dinámicos y funcionales.

Por ejemplo, mencione cómo estos conceptos se aplican en un videojuego sencillo:

- En un juego de aventuras, las **variables** pueden guardar información como los puntos del jugador o su nivel de energía.
- Los **bucles** permiten que el juego actualice constantemente la posición de los personajes o enemigos.
- Los **condicionales** determinan qué sucede si la jugadora o el jugador chocan con un obstáculo, recogen un premio o alcanzan el objetivo.

Explique que los condicionales son esenciales para que los programas tomen decisiones. Por ejemplo, un condicional podría hacer que el personaje principal de un juego pierda puntos si toca una trampa o gane puntos si recoge una estrella, dependiendo de las acciones de la persona que está jugando. Esto permite que el programa sea más interactivo y flexible.

Si cuenta con tiempo, pida a sus estudiantes que en los mismos grupos de trabajo creen ejemplos, similares a los presentados anteriormente, sobre cómo se pueden aplicar los bucles, las variables y los condicionales en otros aspectos de la vida, como los deportes, las actividades de la casa, el baile, el dibujo, entre otros. Una vez discutan sus ejemplos, puede seleccionar algunos para compartir con la clase.

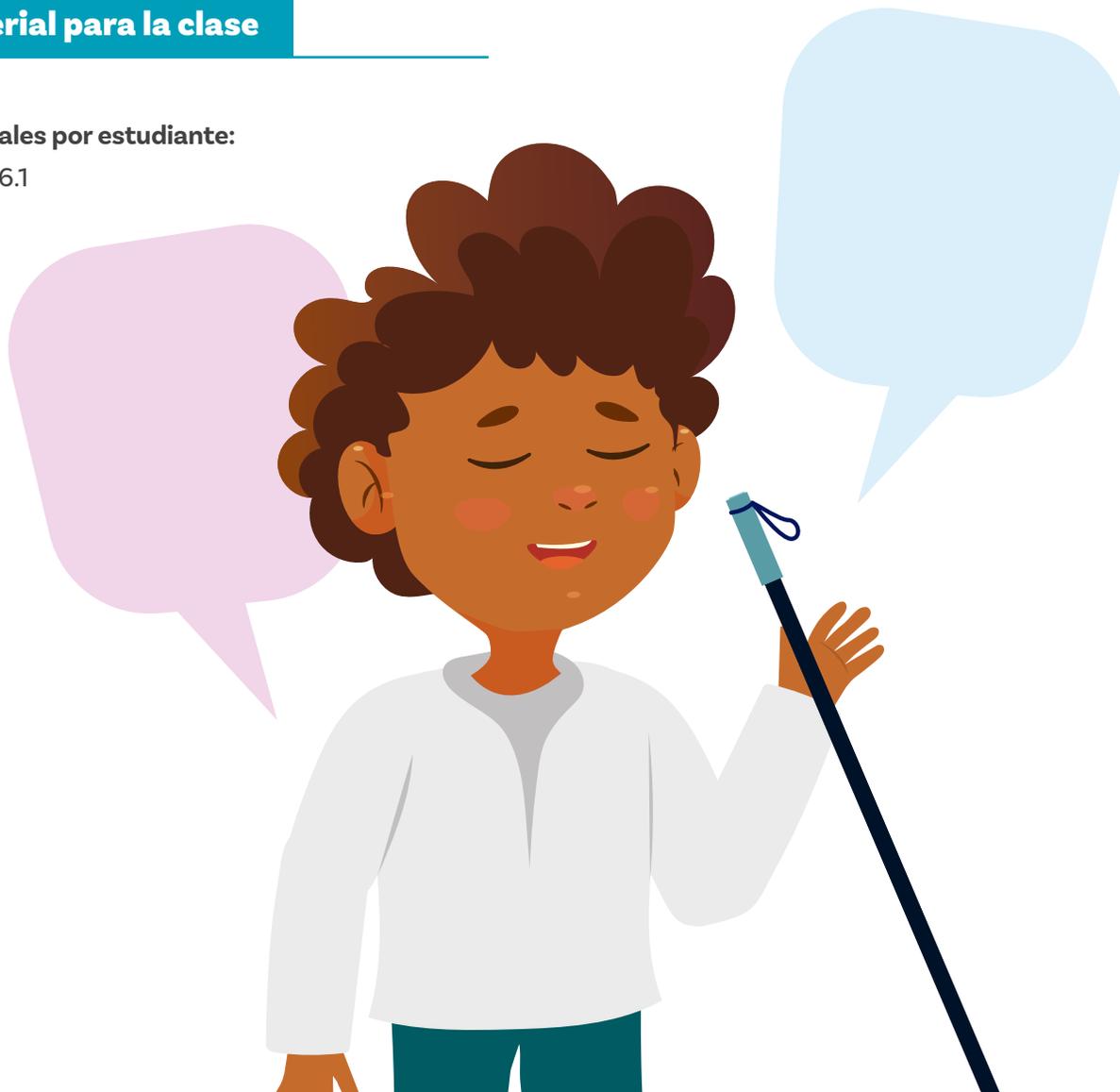
Para finalizar, complementa con sus estudiantes el gráfico de anclaje iniciado en sesiones anteriores.

# Sesión

# 6

## Material para la clase

- Materiales por estudiante:  
Anexo 6.1



## Anexo

### Anexo 6.1

Nombre: \_\_\_\_\_ Fecha: \_\_\_\_\_

**Instrucciones:** Responde cada pregunta o completa la actividad en Scratch según se indica. Recuerda utilizar tus conocimientos sobre bucles y condicionales para resolver los problemas.

**Definición de Términos (2 puntos)**  
 Define los siguientes términos relacionados con la programación en tus propias palabras:

Bucle:

\_\_\_\_\_

\_\_\_\_\_

Variable:

\_\_\_\_\_

\_\_\_\_\_

**Algoritmos Simples (3 puntos)**  
 Escribe un algoritmo paso a paso para resolver cada uno de los siguientes problemas:

- Contar del 1 al 30 utilizando un bucle.
- Crear un programa que cuente de cinco en cinco hasta llegar a 50.

## Evaluación

En esta sesión se invita a sus estudiantes a realizar una evaluación de cierre de la guía pedagógica. Empiece la clase revisando con sus estudiantes lo que han aprendido en las semanas previas. Use los registros o memorias construidas en cada sesión, donde se resumieron los aprendizajes identificados por sus estudiantes.

## Manos a la obra

Entregue las copias necesarias del Anexo 6.1 y lea los enunciados en voz alta con la clase. Luego, dé tiempo a sus estudiantes para completar el cuestionario de forma independiente.

## Nota

En caso de que requiera realizar adaptaciones en la evaluación para estudiantes con aprendizajes diferenciados o para personas con discapacidades, como visuales, puede plantear a sus estudiantes actividades para evaluar que sean diversas y se ajusten a distintas habilidades. Por ejemplo, un proyecto práctico en grupo utilizando el sistema de pares amigos o una presentación oral. De igual manera puede considerar elementos como el uso de tecnologías asistivas o la flexibilización del tiempo.

## Antes de irnos

La siguiente lista de cotejo puede ser usada como guía:

Aprendizajes	
Sus estudiantes se involucran en la actividad.	
Sus estudiantes definen los términos según lo visto en clase.	
Sus estudiantes solucionan los retos propuestos a través de algoritmos simples.	
Sus estudiantes logran modificar el programa en Scratch para incluir bucles.	
Sus estudiantes logran modificar el programa en Scratch para incluir variables.	
Sus estudiantes explican sus soluciones.	

Anexo 1.1 Fichas anotaciones 1

<p><b>Fichas anotaciones 1</b></p> <hr/> <hr/> <hr/> <hr/>	<p><b>Fichas anotaciones 1</b></p> <hr/> <hr/> <hr/> <hr/>
<p><b>Fichas anotaciones 1</b></p> <hr/> <hr/> <hr/> <hr/>	<p><b>Fichas anotaciones 1</b></p> <hr/> <hr/> <hr/> <hr/>
<p><b>Fichas anotaciones 1</b></p> <hr/> <hr/> <hr/> <hr/>	<p><b>Fichas anotaciones 1</b></p> <hr/> <hr/> <hr/> <hr/>
<p><b>Fichas anotaciones 1</b></p> <hr/> <hr/> <hr/> <hr/>	<p><b>Fichas anotaciones 1</b></p> <hr/> <hr/> <hr/> <hr/>

Anexo 1.2 Fichas anotaciones 2

<p><b>Fichas anotaciones 2</b></p> <hr/> <hr/> <hr/> <hr/>	<p><b>Fichas anotaciones 2</b></p> <hr/> <hr/> <hr/> <hr/>
<p><b>Fichas anotaciones 2</b></p> <hr/> <hr/> <hr/> <hr/>	<p><b>Fichas anotaciones 2</b></p> <hr/> <hr/> <hr/> <hr/>
<p><b>Fichas anotaciones 2</b></p> <hr/> <hr/> <hr/> <hr/>	<p><b>Fichas anotaciones 2</b></p> <hr/> <hr/> <hr/> <hr/>
<p><b>Fichas anotaciones 2</b></p> <hr/> <hr/> <hr/> <hr/>	<p><b>Fichas anotaciones 2</b></p> <hr/> <hr/> <hr/> <hr/>

Anexo 3.1 Fichas de estudio 1

al hacer clic en  bajar lápiz

mover 40 pasos

girar 72 grados

Fichas de estudio 1

al hacer clic en  bajar lápiz

mover 40 pasos

girar 72 grados

Fichas de estudio 1

al hacer clic en  bajar lápiz

mover 40 pasos

girar 72 grados

Fichas de estudio 1

al hacer clic en  bajar lápiz

mover 40 pasos

girar 72 grados

Fichas de estudio 1

al hacer clic en  bajar lápiz

mover 40 pasos

girar 72 grados

Fichas de estudio 1

al hacer clic en  bajar lápiz

mover 40 pasos

girar 72 grados

Fichas de estudio 1

al hacer clic en  bajar lápiz

mover 40 pasos

girar 72 grados

Fichas de estudio 1

al hacer clic en  bajar lápiz

mover 40 pasos

girar 72 grados

Fichas de estudio 1

Anexo 3.2 Fichas de estudio 2



Fichas de estudio 2



Fichas de estudio 2



Fichas de estudio 2



Fichas de estudio 2



Fichas de estudio 2



Fichas de estudio 2



Fichas de estudio 2

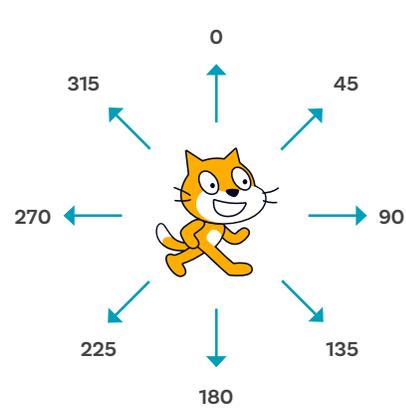
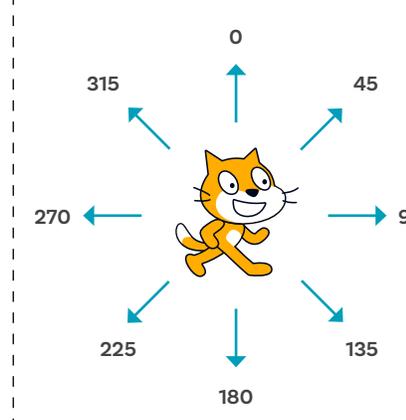
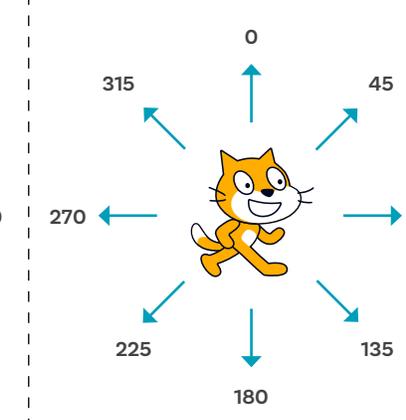
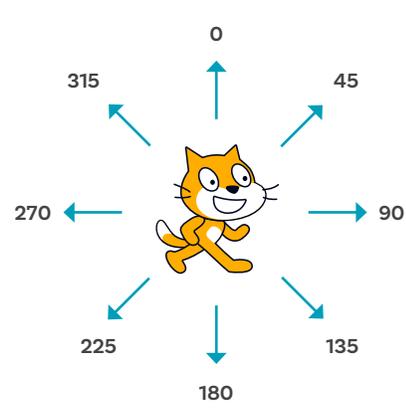
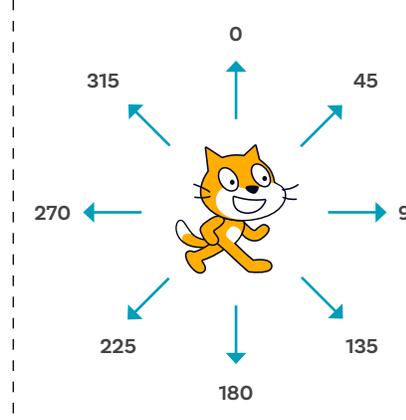
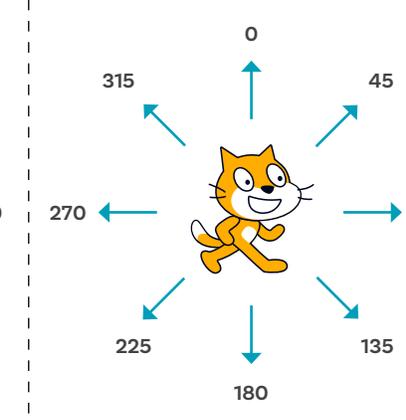
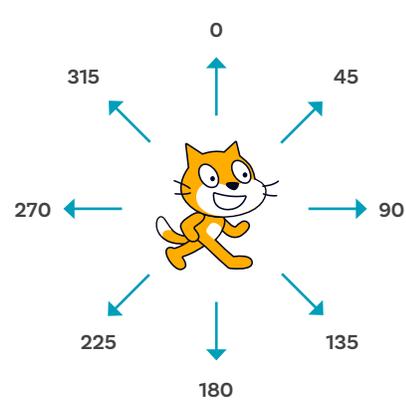
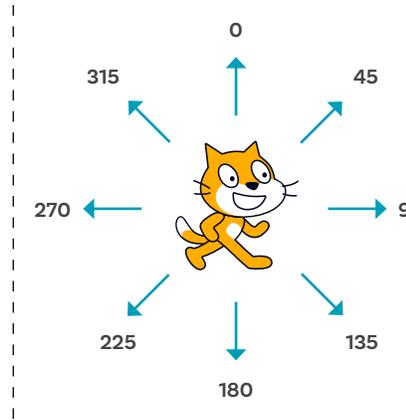
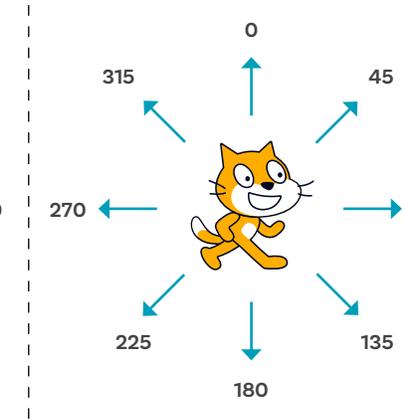


Fichas de estudio 2



Fichas de estudio 2

Anexo 3.3 Fichas de direcciones

 <p>Fichas de direcciones</p>	 <p>Fichas de direcciones</p>	 <p>Fichas de direcciones</p>
 <p>Fichas de direcciones</p>	 <p>Fichas de direcciones</p>	 <p>Fichas de direcciones</p>
 <p>Fichas de direcciones</p>	 <p>Fichas de direcciones</p>	 <p>Fichas de direcciones</p>

## Anexo 3.4 Fichas anotaciones 3

**Fichas anotaciones 3**

¿Qué notas de diferentes en este programa?

---

---

---

Piensa en una idea de programa que puede realizarse de esta forma y escribela.

---

---

---

**Fichas anotaciones 3**

¿Qué notas de diferentes en este programa?

---

---

---

Piensa en una idea de programa que puede realizarse de esta forma y escribela.

---

---

---

**Fichas anotaciones 3**

¿Qué notas de diferentes en este programa?

---

---

---

Piensa en una idea de programa que puede realizarse de esta forma y escribela.

---

---

---

**Fichas anotaciones 3**

¿Qué notas de diferentes en este programa?

---

---

---

Piensa en una idea de programa que puede realizarse de esta forma y escribela.

---

---

---

**Fichas anotaciones 3**

¿Qué notas de diferentes en este programa?

---

---

---

Piensa en una idea de programa que puede realizarse de esta forma y escribela.

---

---

---

**Fichas anotaciones 3**

¿Qué notas de diferentes en este programa?

---

---

---

Piensa en una idea de programa que puede realizarse de esta forma y escribela.

---

---

---

Anexo 3.5 Guía de trabajo 1

**Guía de trabajo 1**

Analicen el programa entregado y deduzcan cuál es su función. Escriban sus notas al respecto.

---



---



---

Ahora, repliquen el programa en Scratch y observen su funcionamiento.

¿Coincide con lo que habían pensado?

¿En qué se diferencia?

Usando como modelo este programa, su reto es crear un programa que permita determinar el promedio de las edades de un grupo de estudiantes de un curso. Recuerden preguntar cuántos estudiantes hay en el grupo.

**Guía de trabajo 1**

Analicen el programa entregado y deduzcan cuál es su función. Escriban sus notas al respecto.

---



---



---

Ahora, repliquen el programa en Scratch y observen su funcionamiento.

¿Coincide con lo que habían pensado?

¿En qué se diferencia?

Usando como modelo este programa, su reto es crear un programa que permita determinar el promedio de las edades de un grupo de estudiantes de un curso. Recuerden preguntar cuántos estudiantes hay en el grupo.

**Guía de trabajo 1**

Analicen el programa entregado y deduzcan cuál es su función. Escriban sus notas al respecto.

---



---



---

Ahora, repliquen el programa en Scratch y observen su funcionamiento.

¿Coincide con lo que habían pensado?

¿En qué se diferencia?

Usando como modelo este programa, su reto es crear un programa que permita determinar el promedio de las edades de un grupo de estudiantes de un curso. Recuerden preguntar cuántos estudiantes hay en el grupo.

**Anexo 4.1** Fichas anotaciones 4

<p><b>Fichas anotaciones 4</b></p> <hr/> <hr/> <hr/> <hr/>	<p><b>Fichas anotaciones 4</b></p> <hr/> <hr/> <hr/> <hr/>
<p><b>Fichas anotaciones 4</b></p> <hr/> <hr/> <hr/> <hr/>	<p><b>Fichas anotaciones 4</b></p> <hr/> <hr/> <hr/> <hr/>
<p><b>Fichas anotaciones 4</b></p> <hr/> <hr/> <hr/> <hr/>	<p><b>Fichas anotaciones 4</b></p> <hr/> <hr/> <hr/> <hr/>
<p><b>Fichas anotaciones 4</b></p> <hr/> <hr/> <hr/> <hr/>	<p><b>Fichas anotaciones 4</b></p> <hr/> <hr/> <hr/> <hr/>

**Anexo 6.1** Evaluación de las actividades

Nombre: \_\_\_\_\_ Fecha: \_\_\_\_\_

**Instrucciones:** Responde cada pregunta o completa la actividad en *Scratch* según se indica. Recuerda utilizar tus conocimientos sobre bucles y condicionales para resolver los problemas.

**Definición de términos (2 puntos):**

Define los siguientes términos relacionados con la programación en tus propias palabras:

Bucle:

---

---

---

---

Variable:

---

---

---

---

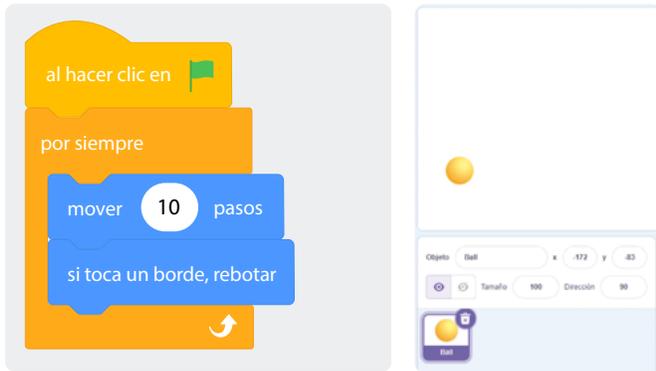
**Algoritmos simples (3 puntos):**

Escribe un algoritmo paso a paso para resolver cada uno de los siguientes problemas:

- a. Contar del 1 al 10 utilizando un bucle.
- b. Crear un programa que cuente de cinco en cinco hasta llegar a 50.

**Modificación de programas en Scratch (5 puntos):**

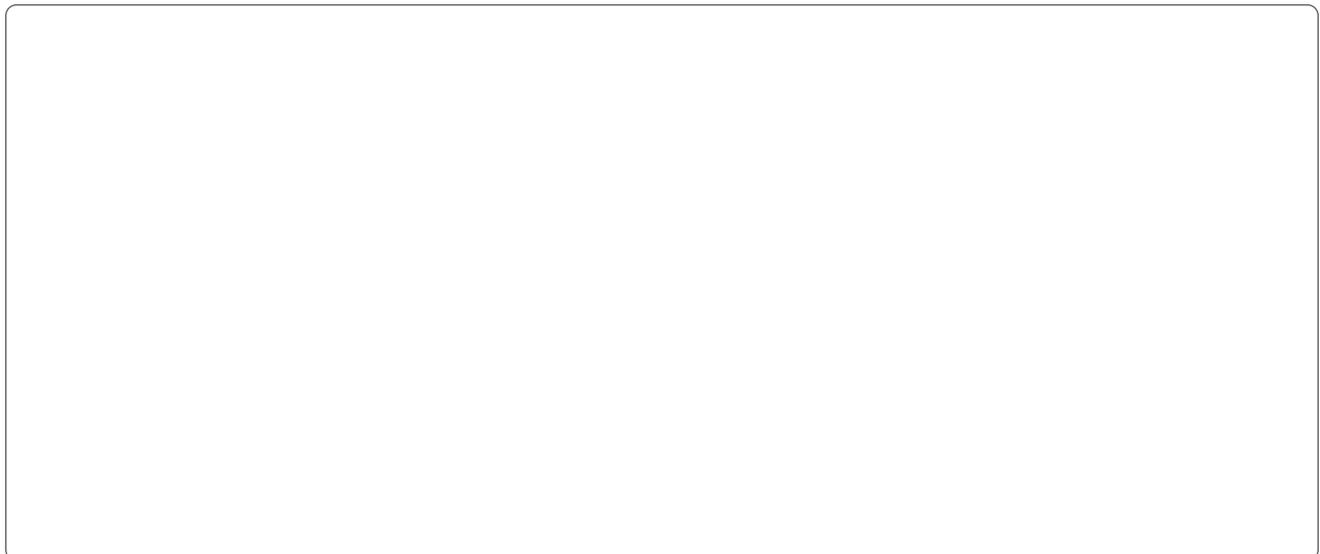
Modifica el programa de Scratch proporcionado para que cumpla con las siguientes especificaciones:



1. El objeto se mueve el doble de rápido.
2. Agrega una condición para que el objeto cambie de color cuando toque un borde de la pantalla.
3. Implementa un contador llamado Cambios que muestre cuántas veces el objeto ha cambiado de color.

**Resolución de problemas en Scratch (5 puntos):**

Simula el comportamiento de un semáforo con luces que cambian de rojo a amarillo, a verde, y viceversa.



**Reflexión y discusión (2 puntos):**

Reflexiona sobre tu experiencia utilizando bucles y condicionales en *Scratch* y responde las siguientes preguntas:

- ¿Cuál fue la parte más desafiante de las actividades?

---

---

---

---

---

---

---

---

---

---

- ¿En qué situaciones crees que podrías aplicar bucles y condicionales fuera de *Scratch*?

---

---

---

---

---

---

---

---

---

---

**Anexo 6.2** Solución a la actividad evaluativa

Nombre: \_\_\_\_\_ Fecha: \_\_\_\_\_

**Instrucciones:** Responde cada pregunta o completa la actividad en Scratch según se indica. Recuerda utilizar tus conocimientos sobre bucles y condicionales para resolver los problemas.

**Definición de términos (2 puntos):**

Define los siguientes términos relacionados con la programación en tus propias palabras:

Bucle:

**Un bucle es una instrucción en programación que permite repetir una acción o un conjunto de acciones varias veces sin tener que escribir el mismo código muchas veces.**

---

---

Variable:

**En Scratch, las variables nos ayudan a guardar números, palabras u otros datos que pueden cambiar mientras el programa se ejecuta.**

---

---

**Algoritmos simples (3 puntos):**

Escribe un algoritmo paso a paso para resolver cada uno de los siguientes problemas:

- a. Contar del 1 al 10 utilizando un bucle.

**Iniciar el programa.**

**Crear una variable llamada número y asignarle el valor inicial de 1.**

**Usar un bucle "Repetir 10" para hacer lo siguiente:**

- **Mostrar el valor de número en la pantalla.**
- **Sumar 1 a la variable**

- b. Crear un programa que cuente de cinco en cinco hasta llegar a 50.

**Iniciar el programa.**

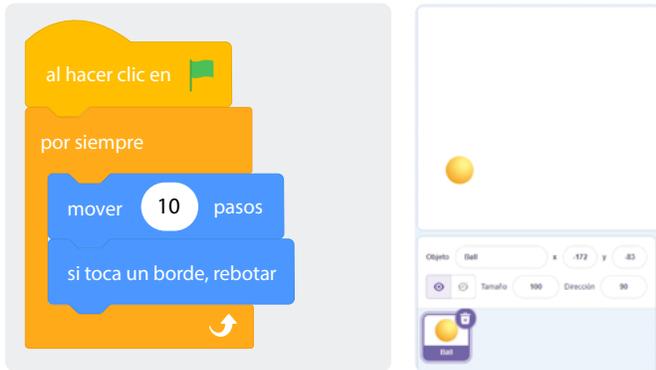
**Crear una variable llamada número y asignarle el valor inicial de 5.**

**Usar un bucle Repetir hasta que número mayor que 50 para hacer lo siguiente:**

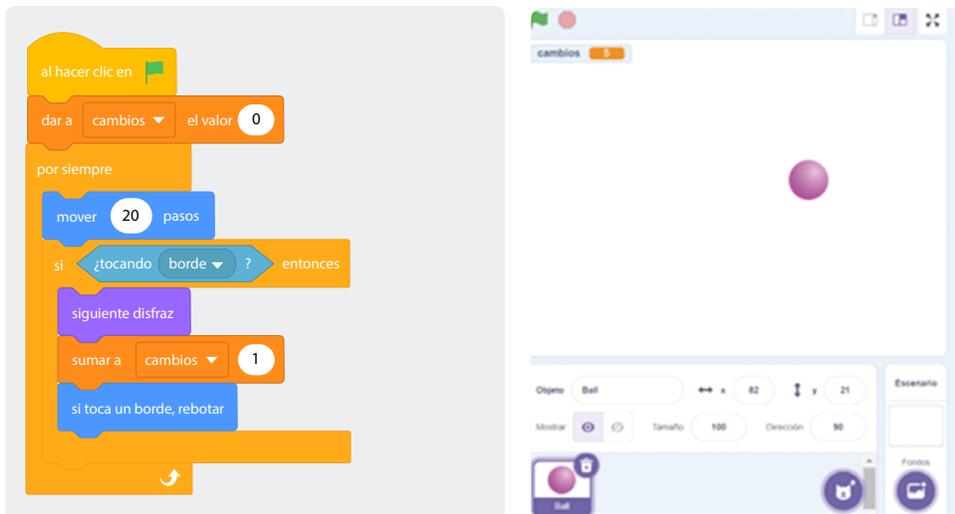
- **Mostrar el valor de número en la pantalla.**
- **Sumar 5 a la variable**

**Modificación de programas en Scratch (5 puntos):**

Modifica el programa de Scratch proporcionado para que cumpla con las siguientes especificaciones:

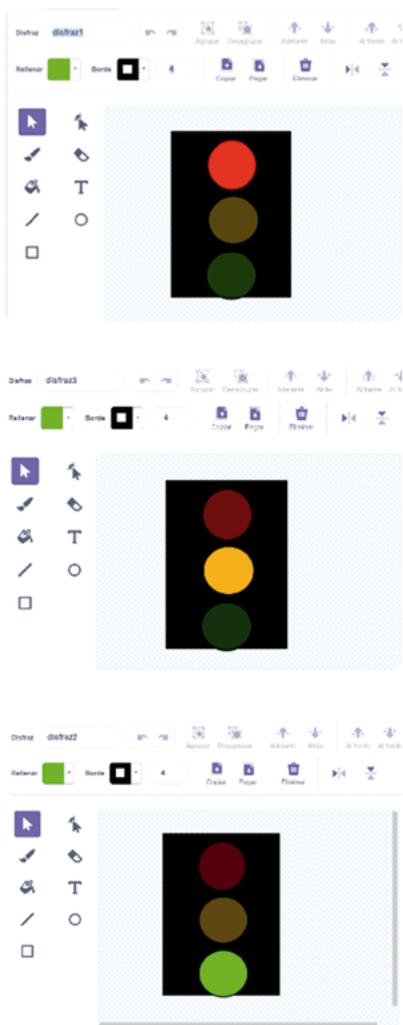


1. El objeto se mueve el doble de rápido.
2. Agrega una condición para que el objeto cambie de color cuando toque un borde de la pantalla.
3. Implementa un contador llamado Cambios que muestre cuántas veces el objeto ha cambiado de color.

**Resolución de problemas en Scratch (5 puntos):**

Simula el comportamiento de un semáforo con luces que cambian de rojo a amarillo, a verde, y viceversa.

**Nota para quien enseña:** A continuación, se presenta una idea sobre cómo abordar el reto. Sus estudiantes podrían elegir simular el semáforo con objetos existentes de Scratch.



**Reflexión y discusión (2 puntos):**

Reflexiona sobre tu experiencia utilizando bucles y condicionales en *Scratch* y responde las siguientes preguntas:

- ¿Cuál fue la parte más desafiante de las actividades?
- ¿En qué situaciones crees que podrías aplicar bucles y condicionales fuera de *Scratch*?

**Respuesta libre**



# TIC



Apoya:



**Educación**

