¿Cómo se propagan los virus?

Grado 6º Guía 2

TIC



Estudiantes

Apoya:

Educación

BRITISH COUNCIL

Colombia Programa
{EL CÓDIGO A TU FUTURO}

¿Cómo se propagan los virus?

Grado 6º Guía 2

TIC

Estudiantes

Apoya:

Educación

BRITISH COUNCIL

Colombia Programa
{EL CÓDIGO A TU FUTURO}

MINISTERIO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

Yeimi Carina Murcia Yela
Ministra TIC

Giovanny Andrés López Cabezas
Viceministro de Transformación Digital

Gloria Patricia Perdomo Rangel
Viceministra de Conectividad

William Alexander Sánchez Martínez
Director (e) de Apropiación de TIC

Lorena Paola Fortich Tulena
Jefe de la Oficina Asesora de Prensa

Equipo Comunicaciones DATIC
Manuel Alejandro Guerrero Aponte
Carlos Andrés Méndez Silva

Equipo Técnico
Cristhiam Fernando Jácome Jiménez
Miller Jimmy Alarcón
Ricardo Cañón Moreno

BRITISH COUNCIL COLOMBIA

Felipe Villar Stein
Director de país

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

Marianella Ortiz Montes
Jefe de Colegios

Óscar Daniel Barrios Díaz
Jefe de Implementación Colombia Programa

Equipo operativo
Juanita Camila Ruiz Díaz
Jesica Lubrano Lavadera
Dayra Maritza Paz Calderón
Saúl Fabián Torres
César Augusto Herrera Lozano
Myriam Nathalia Salinas Tarazona
Paula Álvarez Peña
Elizabeth Marín Giraldo

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

Edición y revisión técnica de versiones finales
Alejandro Espinal Duque
Ana Lorena Molina Castro
Heidy Esperanza Gordillo Bogota
Juan Camilo Londoño Estrada

Apoyo en la edición y revisión académica y técnica (versión 1)
Vanesa Abad Rendón
Raisa Marcela Ortiz Cardona

Consultora experta
Heidy Esperanza Gordillo Bogota

Edición
Juanita Camila Ruiz Díaz
Alexandra Ruiz Correa

BRITISH COMPUTER SOCIETY (BCS) – Consultoría internacional

Julia Adamson
Directora General de Educación y Beneficio Público

Niel McLean
Jefe de Educación

Claire Williams
Coordinadora de Alianzas

ASOCIACIÓN COLOMBIANA 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
Alex González Jurado
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 Ramo

.Puntoaparte Editores

Diseño, diagramación, ilustración, y revisión de estilo

CC BY-NC 4.0

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

"La presente guía corresponde a una versión modificada del material originalmente diseñado por el equipo de autores de ACOFI, en tanto incorpora ajustes, actualizaciones y contribuciones posteriores realizadas por el equipo del British Council y el Ministerio de Tecnologías de la Información y las Comunicaciones, bajo la autorización correspondiente, para su publicación final".

ISBN: 978-628-7934-74-0

Prólogo

Ministerio de Tecnologías de la Información y las Comunicaciones

Estimadas y estimados educadores, estudiantes y comunidad educativa:

Colombia es un país de talento y de oportunidades que no siempre han estado al alcance de todas y todos. Entre ellas, la tecnología, que en las zonas más apartadas aún se percibe como un privilegio lejano.

Con la convicción de conectar territorios y cerrar brechas, emprendimos una transformación en el Ministerio TIC para acercarlo más a las regiones y a sus realidades.

Creemos en una tecnología que llegue con propósito a la vida de las personas, que permita cerrar brechas, abrir caminos y acompañar el desarrollo de las comunidades en todo el país.

Hoy, en ese camino, en el que ya hemos formado a cerca de un millón de colombianas y colombianos en habilidades digitales, presentamos las **Guías de Pensamiento Computacional de Colombia Programa**, una apuesta concreta para llevar estas oportunidades a las aulas del país.

Este material es el resultado de un trabajo colectivo en el que más de 2.000 docentes de 420 sedes educativas aportaron su experiencia y conocimiento para su construcción.

Además, contó con la asesoría de la *British Computer Society (BCS)*, entidad del Reino Unido con más de 20 años de experiencia en la incorporación del pensamiento computacional en el currículo educativo, gracias al convenio de cooperación internacional suscrito con el British Council Colombia.

Estas guías trazan una ruta pedagógica para que niñas, niños y jóvenes desarrollen nuevas formas de pensar, aprendan a resolver problemas y construyan soluciones. Son un recurso pensado para acompañar la labor de los y las docentes que enseñan pensamiento computacional.

Su uso requiere la mirada y la adaptación de quienes enseñan, teniendo en cuenta el contexto y las particularidades de cada estudiante.

Con ellas buscamos que las nuevas generaciones pasen de ser usuarias de la tecnología a protagonistas de su creación y transformación; que puedan comprenderla, dominarla y ponerla al servicio de sus propias vidas y de sus comunidades.

Queremos que, sin importar el lugar donde se estudie, en una zona rural, en la Amazonía o en una ciudad, existan las mismas oportunidades para desarrollar talento digital.

El gobierno del presidente Petro le cumplió a los colombianos y las colombianas y, por eso, buscamos que más niñas y jóvenes se reconozcan como creadoras, y que cada docente cuente con herramientas para transformar su aula.

Porque cuando conectamos territorios, abrimos oportunidades para que Colombia avance con equidad, desarrollo y justicia social.

Invitamos a estudiantes, docentes y comunidades a apropiarse de estas guías, a disfrutarlas y a llevarlas a su vida diaria. Porque cuando la educación digital llega a los territorios, no solo transforma el aprendizaje: transforma comunidades enteras y abre caminos para construir un mejor país.

Carina Murcia Yela

Carina Murcia Yela
Ministra de Tecnologías de la Información y las Comunicaciones
Gobierno de Colombia

Prólogo

British Council

Por más de 87 años, el *British Council* ha tenido presencia en el país, buscando construir lazos de confianza y cooperación entre el Reino Unido y Colombia. Uno de los mejores ejemplos de lo que podemos lograr a través de estas alianzas es *Colombia Programa*. Esta iniciativa, que desde hace tres años y de la mano con el Ministerio de Tecnologías de la Información y las Comunicaciones nos ha permitido alcanzar un objetivo ambicioso: ofrecer espacios de formación y recursos de alta calidad para que docentes y estudiantes de los colegios públicos del país desarrollen sus habilidades digitales y el pensamiento computacional con un enfoque de género.

Esta publicación marca un hito particularmente importante, pues con ella entregamos al país un material sin precedentes en América Latina y comparable con muy pocos a nivel mundial. Se trata de 79 guías pedagógicas que suman más de 4.000 páginas y contienen 30 sesiones por cada grado, desde transición hasta undécimo. Con estas, maestras y maestros podrán facilitar al menos una hora de clase semanal orientada al desarrollo del pensamiento computacional de sus estudiantes. Basta preguntarse qué otra área o asignatura en Colombia cuenta con un material semejante —en términos de amplitud, rigurosidad y acceso gratuito— para dimensionar la relevancia de este esfuerzo para el sector educativo y tecnológico del país.

Al abrir estas páginas impresas o navegar su versión digital, encontrarán una gran variedad de actividades que proponen a niños, niñas y jóvenes nuevas formas de acercarse a los temas más relevantes del presente, como el uso de la inteligencia artificial desde un enfoque ético y seguro. También verán cómo esta y otras temáticas se articulan con diversas áreas del conocimiento, promoviendo el trabajo colaborativo entre docentes y estudiantes y convirtiendo el pensamiento computacional en un puente entre las matemáticas, las ciencias, el arte y el lenguaje.

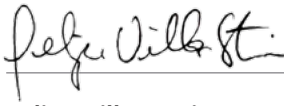
Encontrarán también en estas páginas orientaciones y sugerencias sobre cómo adaptar las actividades a diferentes contextos y necesidades. Por ejemplo, cómo desarrollar habilidades de pensamiento computacional en instituciones sin conectividad o incluso sin electricidad; cómo fomentar la equidad de género para que más niñas y jóvenes se interesen por la programación y las áreas STEM; o cómo ajustar ciertas actividades para que estudiantes en condición de discapacidad puedan participar plenamente.

Lo que quizá no sea visible a primera vista es el arduo proceso que hizo posible llegar a este punto. Un equipo de 25 personas se dedicó durante más de un año a diseñar y revisar cada una de las 79 guías. Contamos, además, con la asesoría de la *British Computer Society (BCS)*, a través de expertos que participaron y lideraron reformas curriculares en Inglaterra en el área de ciencias de la computación. Asimismo, realizamos 38 mesas de validación en cinco regiones del país, en las que 1.755 docentes revisaron y comentaron los primeros borradores. A esto se suman las innumerables horas que más de 2.100 docentes "pares expertos", pertenecientes a los 420 Nodos de Desarrollo de Pensamiento Computacional en todo el país, dedicaron en 2025 a pilotear las guías en sus aulas, asegurando así su pertinencia para los contextos educativos reales.

A cada uno de estos maestros y maestras, directivas docentes, mentores(as), tutores, expertos(as), académicos(as) y funcionarios(as) del *British Council* y de los Ministerios de las TIC y de Educación, así como a los y las estudiantes que nos acompañaron en este proceso, infinitas **gracias** por permitir que hoy Colombia cuente con esta herramienta tan poderosa.

Desde el *British Council*, sabemos que estas guías de pensamiento computacional abrirán nuevas posibilidades para que, a través del trabajo y esfuerzo de docentes en el aula, los y las estudiantes accedan a nuevas formas de pensar, de interactuar con la tecnología y de desarrollar capacidades alineadas con las demandas del mercado laboral presente y futuro. De lograrlo, conseguiremos el sueño que ha guiado esta labor y la alianza de *Colombia Programa*: que cada niña, niño y joven en Colombia pueda construir el futuro digital y material que desee.

Esperamos que las disfruten y que juntos podamos aprovechar al máximo su potencial.



Felipe Villar Stein
Director
British Council Colombia

El camino recorrido para llegar a estas guías

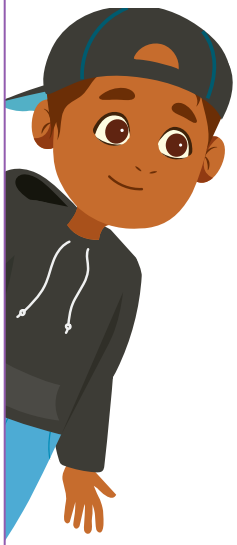
Creación

1 **8 de noviembre de 2023**
El Ministerio TIC y el *British Council* firman el convenio 1247 que da inicio a Colombia Programa.

Diciembre de 2023
La Asociación Colombiana de Facultades de Ingeniería (ACOFI) empieza a trabajar en la matriz de alcance y secuencia.

2 **Febrero de 2024**
Se aprueba la matriz de alcance y secuencia de las guías creadas por ACOFI, que incorpora retroalimentación de la *British Computer Society*. Con esta aprobación se inicia la creación de las guías de pensamiento computacional de Colombia Programa.

Validación y pilotaje



3 **Mayo a octubre 2024**
1.755 docentes en 38 mesas de trabajo de las 5 regiones del país, validan la versión inicial de las guías y proponen ajustes para hacerlas más adaptables e inclusivas.

Mayo a diciembre 2024
Se incorporan a las guías los ajustes provenientes de las mesas de validación.

4 **Agosto 2024 a mayo 2025**
Diagramación e impresión de las guías para entrega a los nodos de pensamiento computacional.

Marzo a septiembre 2025
2.100 docentes de los 420 nodos prueban las guías en sus aulas. Se recopilan sugerencias de ajustes a partir de este ejercicio.

Octubre a noviembre 2025
Análisis de resultados y recomendaciones para ajustes de la versión final de las guías.

Ajustes finales



5 **Diciembre 2025 a abril 2026**
Implementación de ajustes finales.

Enero a junio 2026
Diagramación de la versión final de las guías de pensamiento computacional de Colombia Programa.



Grado 6° Guía 2

TIC

Identificación de los aprendizajes

- Lógica, programación y depuración
- Modelación y simulación

Aprendizajes de la guía

Con las actividades de esta guía se espera que puedas avanzar en:

- Crear programas sencillos utilizando un lenguaje de programación en *Scratch*.
- Reconocer un programa como una secuencia de comandos para realizar una tarea que tiene un inicio, eventos, metas y resultados esperados.
- Programar simulaciones sencillas de fenómenos naturales.

Resumen de la guía

En esta guía de 5 lecciones, tienes la oportunidad de profundizar en el lenguaje de programación en un programa llamado *Scratch*. A lo largo de las sesiones, se explora la creación de proyectos interactivos mediante actividades conectadas, con las cuales se pueden identificar las bases fundamentales del lenguaje de programación.

Resumen de las sesiones

Sesión 1

Se aprende a identificar los bloques básicos de código como formas diversas para dar instrucciones, siguiendo instrucciones sencillas para expresar movimientos y construir pequeños programas con bloques de instrucciones.

Sesión 2

Se aprende a utilizar nuevos bloques de código, modificar objetos y fondos, construyendo un programa con bloques de instrucciones más completo con acciones detalladas.

Sesión 3

Se aprende a utilizar funciones (bloques personalizados) en *Scratch* para mejorar la organización de los proyectos, por medio de la creación de figuras geométricas sencillas con la extensión de lápiz.

Si se requiere

En esta guía se trabaja con *Scratch*. Si nunca has trabajado con este editor de programas en bloque, se sugiere realizar primero la Guía 1 de grado 3°.

Sesión 4

Se aprende a utilizar todos los bloques y códigos vistos en sesiones anteriores, creando un proyecto de simulación de la propagación de un virus.

Sesión 5

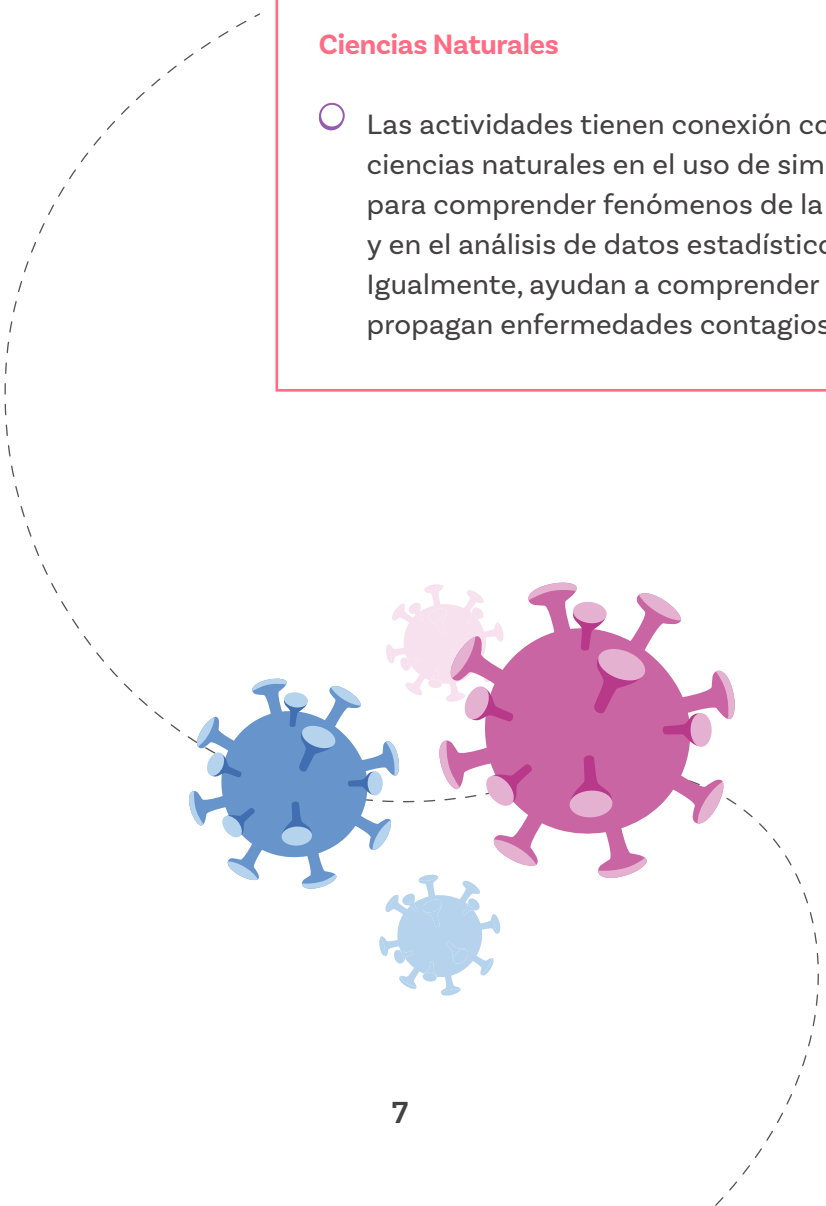
Se avanza un poco más en la utilización de todos los bloques y códigos vistos en sesiones anteriores, creando un proyecto de simulación de la propagación de un virus.

Conexión con otras áreas

A continuación, se presenta la conexión con otras áreas:

Ciencias Naturales

- Las actividades tienen conexión con las ciencias naturales en el uso de simuladores para comprender fenómenos de la naturaleza y en el análisis de datos estadísticos simples. Igualmente, ayudan a comprender cómo se propagan enfermedades contagiosas.



Grado 6° **Guía 2**

TIC

Sesión 1

Aprendizajes esperados

Con las actividades de esta sesión se espera que puedas avanzar en:

- Utilizar diferentes tipos de bloques para crear programas sencillos en *Scratch*.
- Comprender la importancia de la secuencia de instrucciones en los lenguajes de programación en bloques.

Duración sugerida



15% 70% 15%



Material para la sesión

Material por estudiante:

- ○ Anexo 1.1 o una copia de este, visible en una cartelera en el salón.
- ○ Anexo 1.2.
- ○ Dispositivo con acceso a *Scratch*.

Anexo

Anexo 1. 2

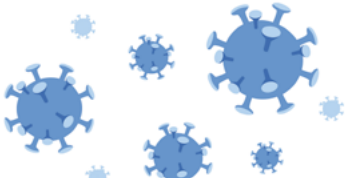
¿Te has preguntado alguna vez cómo se propagan los virus en una comunidad? Las simulaciones son herramientas poderosas que nos permiten entender fenómenos complejos de una manera visual e interactiva. Nos ayudan a prever cómo se comportarían ciertos eventos en la realidad y a tomar decisiones informadas basadas en estos escenarios.

Tu reto en esta guía será programar una simulación en Scratch para entender cómo un virus puede propagarse. La simulación deberá mostrar varias personas sanas en un espacio (pueden ser representadas por círculos de color verde) y una persona enferma (representada en color rojo), todas en posiciones iniciales aleatorias.

Las personas deberán moverse de forma aleatoria y se irán contagiando a medida que entren en contacto con quienes están enfermos. El contagio se mostrará como un cambio de color.

Importante: cuando las personas representadas en la simulación lleguen al borde de la pantalla, deben rebotar y continuar su movimiento.

Para lograrlo deberás utilizar funciones, condicionales, cambios de disfraces, eventos y valores aleatorios para simular el comportamiento del virus, empezando con una sola persona contagiada y observando cómo se propagan los contagios a medida que las personas interactúan.



Lo que sabemos, lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

Empieza por leer el reto que te proponemos en esta guía, el cual encontrarás en el *Anexo 1.2*. El reto lo podrás resolver al final con lo que aprendas en las 5 sesiones.

Los computadores requieren un lenguaje de programación. El lenguaje es una forma en la que le damos instrucciones a un computador para que realice tareas específicas.

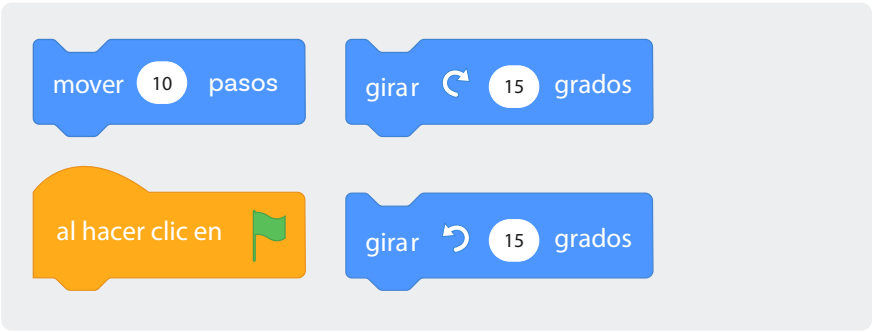
Es probable que en guías anteriores hayas utilizado *MakeCode*, que es un lenguaje de programación basado en bloques para controlar la tarjeta *micro:bit*. Sin embargo, existen muchos lenguajes más y en esta guía aprenderás sobre *Scratch*.

Scratch es un lenguaje de programación visual diseñado para facilitar el aprendizaje de la programación, especialmente para niñas, niños y personas principiantes. Fue desarrollado por el *MIT Media Lab* con el objetivo de permitir a los usuarios crear historias interactivas, juegos y animaciones mediante la combinación de bloques gráficos que representan diferentes comandos y acciones.

Con *Scratch*, el lenguaje de programación se basa en bloques que utilizan secuencias para que los computadores realicen tareas paso a paso.

Observa las imágenes que aparecen en la *Figura 1*.

Figura 1. Bloques básicos de *Scratch*



Nota

Scratch funciona sin conectividad si está instalado en un computador; igualmente, puede accederse desde celulares o tabletas con Android o iOS. Además, este programa puede trabajarse con lectores de pantalla como *NVDA* y *JAWS*. Si necesitas hacer uso de esta característica, pide apoyo a tu docente.



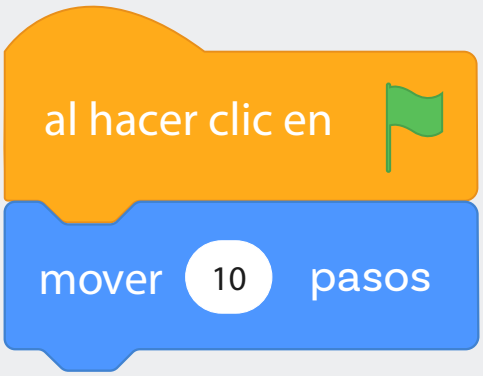
¿Qué se entiende de las imágenes? ¿Para qué crees que se usan esos bloques?

Explica en palabras lo que comprendes de cada una. Luego, comparte con tus compañeras(os) según indique tu docente.

Las imágenes anteriores son bloques de instrucciones de *Scratch*.

Ahora vamos a combinar dos bloques, como se muestra en la *Figura 2*.

Figura 2. Código que combina 2 bloques



¿Qué crees que hará este programa? Explica en palabras.

¿Cuáles bloques puedes combinar para lograr que el programa dirija un objeto para que, además de moverse, gire hacia la derecha 15 grados?

Comparte tus respuestas con tu docente.

Enlace

Acceso a *Scratch*

Manos a la obra

Conectadas

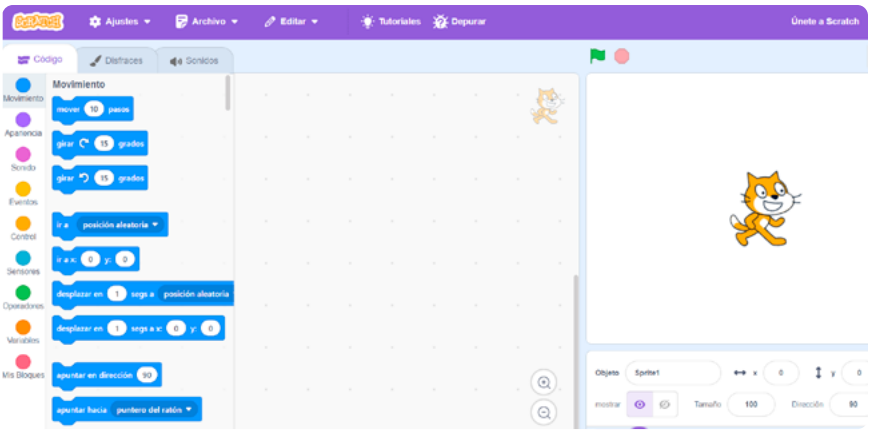
Esta sección corresponde al 85% de avance de la sesión

Organízate en pareja, siguiendo las indicaciones de tu docente. Esta es una buena estrategia en programación, porque al trabajar con otra persona pueden ayudarse mutuamente.

Ahora usen bloques para programar a un gato con el objetivo de que se mueva hacia su casa cuando se haga clic en él. Sigan los pasos que se presentan a continuación y si tienen alguna dificultad, coméntenla con sus compañeras(os) o su docente.

1. Vayan a scratch.mit.edu y hagan clic en **Crear** para iniciar un nuevo proyecto. Al ingresar verán el entorno que se presenta en la *Figura 3.*

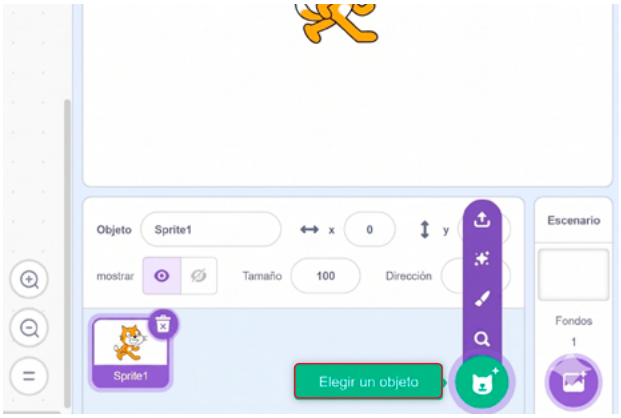
Figura 3. Entorno de programación en *Scratch*



Como pueden ver, *Scratch* tiene un personaje de un gato, que aparece de forma predeterminada.

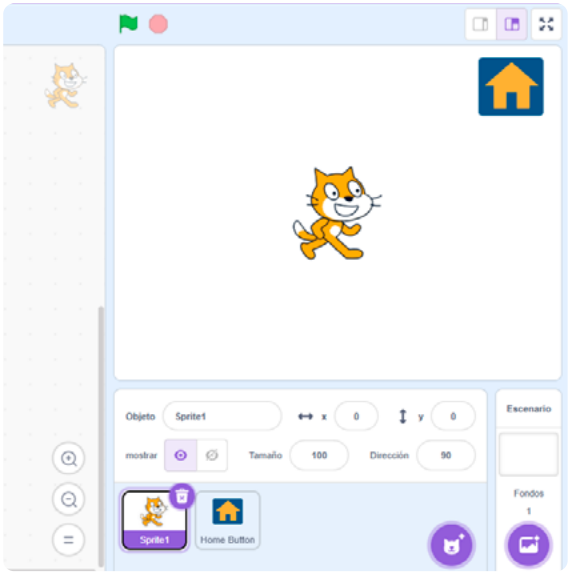
2. En la parte inferior derecha de la interfaz de *Scratch*, hagan clic en el ícono **Elige un objeto**, como se ve en la *Figura 4*, y busquen uno que represente una casa o algún objeto que sea similar.

Figura 4. Comando "Elige un objeto" en *Scratch*



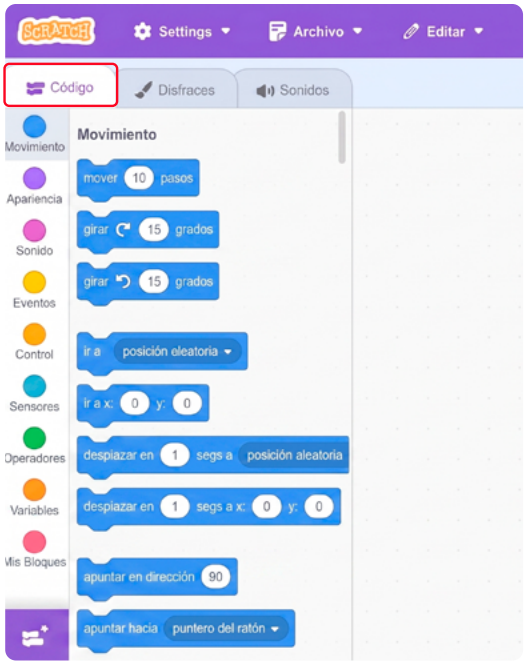
3 Ubiquen al gato y la casa en sus posiciones en el escenario. Asegúrense de que la casa esté en un lugar separado del gato, para que el gato pueda moverse hacia la casa.

Figura 5. Escena de *Scratch* con dos objetos: una casa y el gato



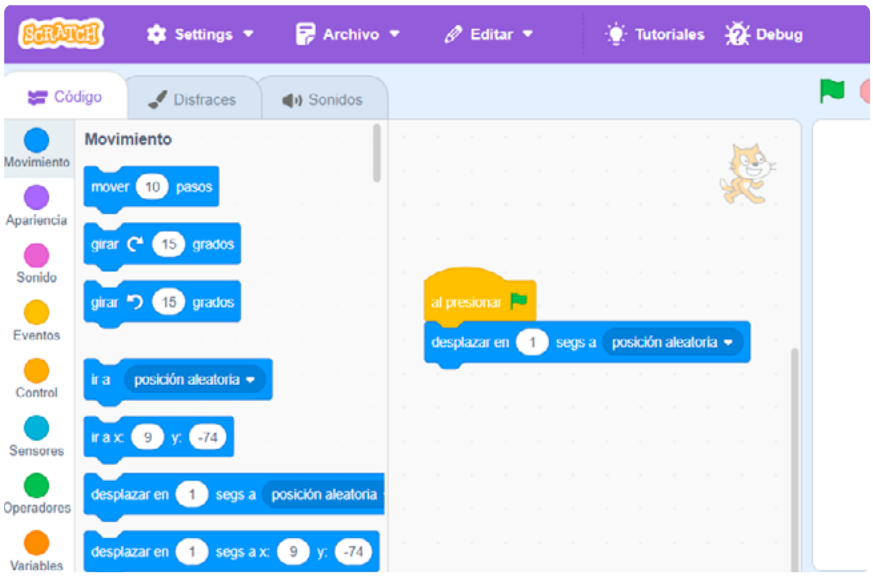
4 Ahora van a programar el gato para que se mueva hacia la casa cuando se haga clic en él. Para hacerlo, seleccionen el gato, y en la parte izquierda, hagan clic en la pestaña **Código**, ver *Figura 6*.

Figura 6. Menú de código en *Scratch*



5 Arrastren y suelten los bloques de código que se observan en la *Figura 7.*

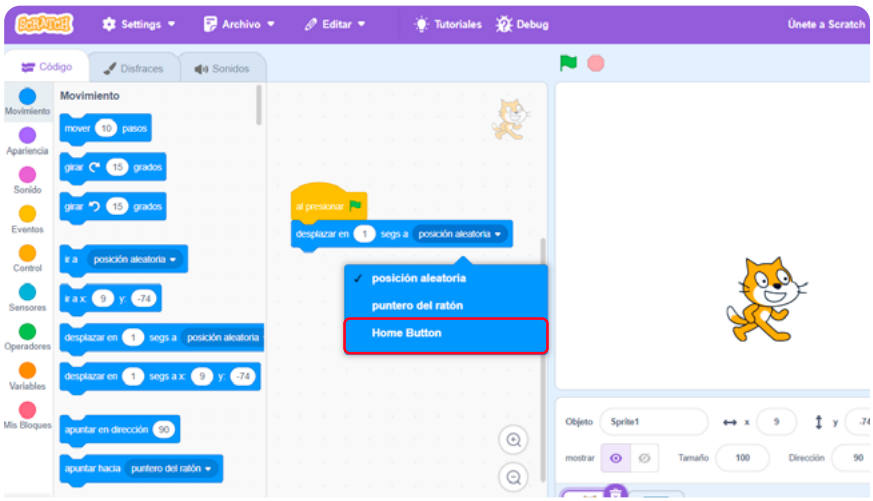
Figura 7. Bloque de eventos y bloque de movimiento





6 Deben dar clic donde señala la flecha y seleccionar el objeto que eligieron previamente, que en el caso de la *Figura 8* se llama **Home Button**.

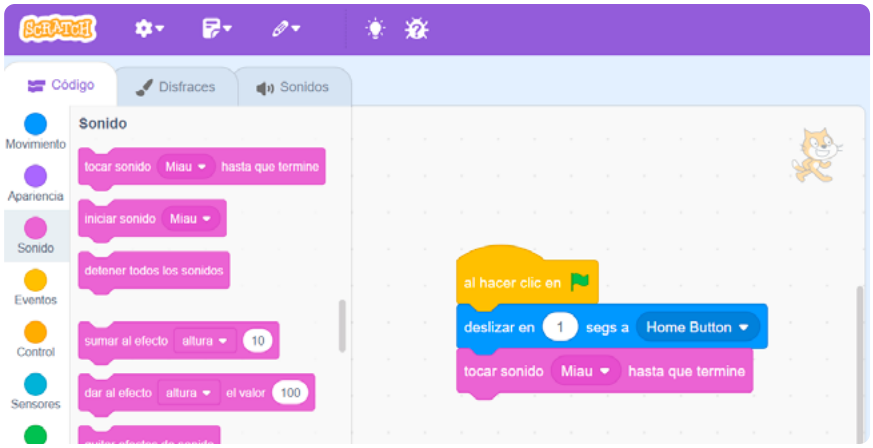
Figura 8. Código de la *Figura 7* y escena descrita en la *Figura 5*



7 Ahora prueben su proyecto haciendo clic en la bandera verde en la parte superior de la pantalla. Esto iniciará la ejecución del programa que han creado.

8 Agreguen sonido que se reproduzca cuando el gato llegue a la casa. Vayan a la pestaña Código y seleccionen Sonido. Escojan el bloque **tocar sonido Miau hasta que termine**, ver *Figura 9*.

Figura 9. Bloque tocar sonido Miau hasta que termine, agregado al final del código de la *Figura 7*



9 Prueben su proyecto haciendo clic en la banderita verde nuevamente.

Como reto adicional, intenten agregar un fondo a su proyecto. ¿Cómo creen que pueden hacerlo? Hagan pruebas y cuando lo logren, compartan con sus compañeras y compañeros la manera en que lo hicieron.

Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

De forma individual, regresa a revisar los aprendizajes esperados. Elige la opción de respuesta que mejor describa lo que alcanzaste.

1 ¿Pudiste fortalecer el uso de diferentes tipos de bloques para crear programas en el lenguaje de bloques *Scratch*?

- ◯ Sí
- ◯ Parcialmente
- ◯ Aún no

2 ¿Comprendes la importancia de la secuencia de instrucciones en los lenguajes de programación en bloques?

- ◯ Sí
- ◯ Parcialmente
- ◯ Aún no

> **Si tus respuestas fueron "Parcialmente" o "Aún no", vuelve a las actividades propuestas en *Scratch*. Luego, discute con tus compañeras y compañeros de grupo lo que se hizo en cada momento de la actividad y lo que iba haciendo cada bloque. Si todavía te quedan dudas, consúltale a tu docente.**

Anexo

Anexo 1.1

Bloque de Scratch	Acción que realiza
al hacer clic en	
tocar sonido Miau hasta que termine	
mover 10 pasos	
deslizar en 1 segs a posición aleatoria	

Anexo 1.2

¿Te has preguntado alguna vez cómo se propagan los virus en una comunidad? Las simulaciones son herramientas poderosas que nos permiten entender fenómenos complejos de una manera visual e interactiva. Nos ayudan a prever cómo se comportarían ciertos eventos en la realidad y a tomar decisiones informadas basadas en estos escenarios.

Tu reto en esta guía será programar una simulación en Scratch para entender cómo un virus puede propagarse. La simulación deberá mostrar varias personas sanas en un espacio (pueden ser representadas por círculos de color verde) y una persona enferma (representada en color rojo), todas en posiciones iniciales aleatorias.

Las personas deberán moverse de forma aleatoria y se irán contagiando a medida que entren en contacto con quienes están enfermos. El contagio se mostrará como un cambio de color.

Importante: cuando las personas representadas en la simulación lleguen al borde de la pantalla, deben rebotar y continuar su movimiento.

Para lograrlo deberás utilizar funciones, condicionales, cambios de disfraces, eventos y valores aleatorios para simular el comportamiento del virus, empezando con una sola persona contagiada y observando cómo se propagan los contagios a medida que las personas interactúan.



Ahora, sigue las instrucciones de tu docente para reunirte en grupos de 4 personas y trabajar una rutina de pensamiento llamada Pensar, Presentar e Integrar (P-P-I):

- Primero responde individualmente.
- Luego, cada persona en su turno le presenta al resto del grupo sus respuestas.
- Finalmente, el grupo integra una respuesta unificada.

Las preguntas que se proponen son las siguientes:

¿Qué otras cosas crees que se pueden programar en *Scratch*?

¿Para qué podrías usar tus nuevos conocimientos? O, si ya tienes experiencia usando el lenguaje, ¿qué podrías mejorarle al programa?

Si regresas al reto del *Anexo 1.2*, ¿para qué serviría lo que has aprendido?

Conversen con sus compañeras y compañeros de grupo sobre estas preguntas. Luego, completen el *Anexo 1.1* para resumir las acciones de las instrucciones que se han utilizado en el proyecto.



Sesión 2

Aprendizajes esperados

Con las actividades de esta sesión se espera que puedas avanzar en:

- Identificar y modificar objetos y sensores en *Scratch*.
- Elaborar un programa que realice acciones complejas mediante el lenguaje de bloques *Scratch*.
- Predecir el resultado de programas sencillos.

Duración sugerida



15% 70% 15%



Material para la sesión

Material por grupo:

- ○ Dispositivo con acceso a *Scratch*.

Lo que sabemos, lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

Como ya sabes, *Scratch* es un lenguaje de programación visual diseñado especialmente para aprender los conceptos básicos de programación de una forma creativa.

En *Scratch*, los objetos son los personajes o elementos que puedes mover y controlar en tus proyectos. Puedes elegir entre muchos objetos que ya están disponibles o crear los tuyos.

El programa también te da la oportunidad de seleccionar un fondo, que es la escena donde ocurren las acciones. Puedes elegir un fondo de la biblioteca de *Scratch*, cargar uno que tú hayas creado o incluso dibujarlo.

Los bloques son las instrucciones que le das a tus objetos. Cada bloque tiene una forma y un color diferente, lo que ayuda a identificar su función. Los bloques que aparecen en la paleta se agrupan en diferentes categorías como las siguientes:

- **Eventos:** indican las acciones que darán inicio al programa o la animación.

  Ejemplo: bloque **al presionar la bandera verde**[1]

Figura 1. Bloque de eventos



¿Para qué sirve? Permite iniciar la animación o ejecución del programa al hacer clic en la bandera verde.

1. Este bloque en algunas versiones de *Scratch* se puede llamar: al hacer clic en la bandera verde.

- **Control:** controla el flujo del programa.
  Ejemplo: bloques **por siempre** y **si ... entonces.**

Figura 2. Bloques por siempre y si... entonces



¿Para qué sirven? Permiten que algunas acciones sucedan continuamente o solo si se cumplen ciertas condiciones.

- **Movimiento:** mueve los objetos en el escenario.
  Ejemplo: bloques **mover (10) pasos** y **girar a la derecha (15) grados.**

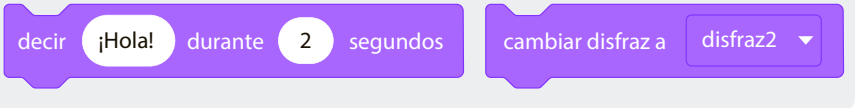
Figura 3. Bloques mover y girar



¿Para qué sirven? Pueden hacer que los objetos se desplacen, giren o vayan a una posición específica.

- **Apariencia:** cambia la forma en que se ve el objeto.
  Ejemplo: bloques **decir [Hola] durante (2) segundos** y **cambiar disfraz a [nombre del disfraz]**.

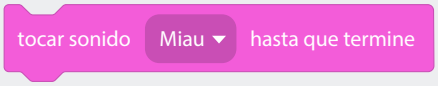
Figura 4. Bloques decir y cambiar disfraz a



¿Para qué sirve? Cambia la apariencia del objeto o permite que este muestre mensajes en la pantalla.

- **Sonido:** agrega efectos de sonido o música a tu proyecto. Ejemplo: bloque **tocar sonido [nombre del sonido].**

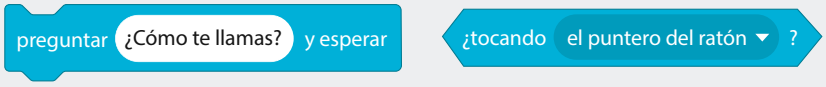
Figura 5. Bloque tocar sonido



¿Para qué sirve? Ejecuta el sonido Miau al ejecutar el proyecto.

- **Sensores:** permite que el objeto interactúe con las y los usuarios. Ejemplo: bloques **¿tocando [el puntero del ratón]?** y **preguntar [¿Cómo te llamas?] y esperar.**

Figura 6. Bloques preguntar y esperando y ¿tocando?

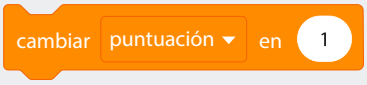


¿Para qué sirve? Genera acciones cuando el objeto toque otro objeto o reciba información del usuario.



- **Variables:** guarda información que puede cambiar durante el juego. Ejemplo: Bloque **cambiar [puntuación] en (1).**

Figura 7. Bloque cambiar



¿Para qué sirve? Incrementa en un punto el valor almacenado en un espacio de memoria identificado con el nombre "puntuación".

Ahora utilizarán algunos de estos códigos y elaborarán un programa para que los objetos interactúen.

Enlace

Scratch

Manos a la obra

Conectadas

Esta sección corresponde al 85% de avance de la sesión

Organízate en pareja, siguiendo las indicaciones de tu docente.

Luego, retomen el *Anexo 1.1* realizado en la sesión anterior. Léanlo y recuerden las acciones que cumplen los bloques que usaron en esa sesión. Esto les ayudará a prepararse para la actividad que van a realizar en esta sesión.

Lean la siguiente descripción de lo que debería hacer el programa que van a crear:

Saray está aprendiendo a programar y quiere darles una sorpresa a sus amigas y amigos el día de su cumpleaños. Para esto, quiere crear una animación que empiece mostrando un personaje que diga "¡Hola!" y pregunte tu nombre. Cuando le respondas, el personaje debe pensar "Mmm" y moverse hasta tocar un pastel de cumpleaños. Al llegar al pastel, el fondo debería cambiar para mostrar una escena de fiesta y el personaje debería aparecer con un sombrero de cumpleaños y decir: "¡Feliz cumpleaños!".

Si prefieren cambiar el motivo de la fiesta por otro que les llame más la atención (ej., un grado, una fiesta sorpresa u otra razón para celebrar), modifiquen el mensaje que dice el personaje Amigo cuando aparezca con el fondo de fiesta.



Figura 8. Ejemplo de escena propuesta



Si tú y tu compañera o compañero tienen experiencia usando *Scratch*, pueden planear el código y trabajar de manera independiente. Si no, sigan los pasos a continuación.

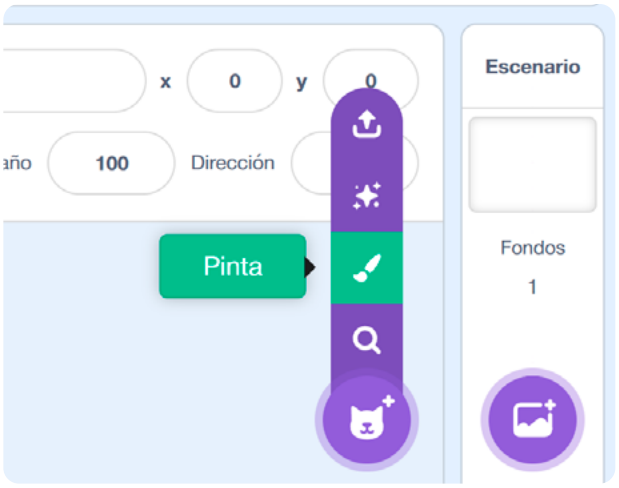
1. Vayan a scratch.mit.edu y hagan clic en “Crear” para iniciar el nuevo proyecto.

2. No usarán el objeto gato en su programa. Elimínenlo dando clic en la caneca que se puede ver en la parte superior derecha del recuadro del personaje. Luego, den clic en el ícono **Elige un objeto** y seleccionen la opción con el pincel llamada **Pinta**, ver *Figuras 9 y 10*.

Figura 9. Menú Elige un objeto

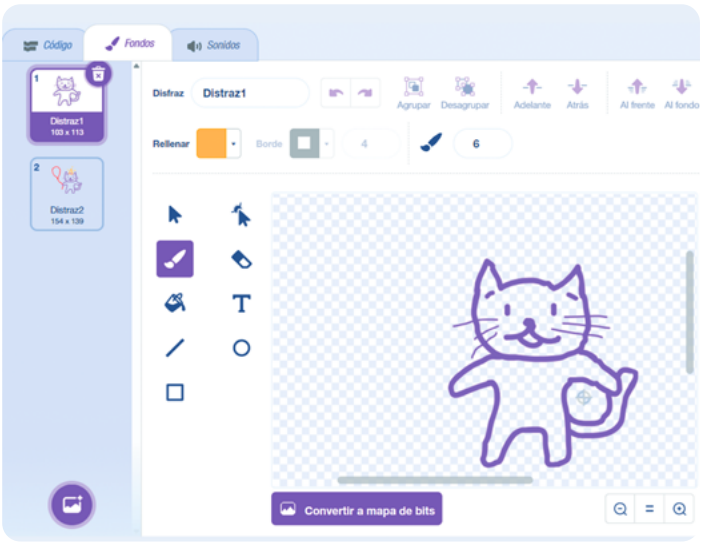


Figura 10. Botón Pinta



3 Ahora mismo se encuentran en la pestaña Disfraces. Ahí encontrarán un tablero donde podrán dibujar su propio objeto. ¡Dibujen un amigo para el gato Scratch! Usen su creatividad y utilicen las herramientas que encuentran al lado izquierdo del tablero, como se muestra en la *Figura 11*.

Figura 11. Ejemplo de personaje creado



4 Cambien el nombre del objeto a “Amigo”, como se muestra en la *Figura 12*.

Figura 12. Propiedades del objeto

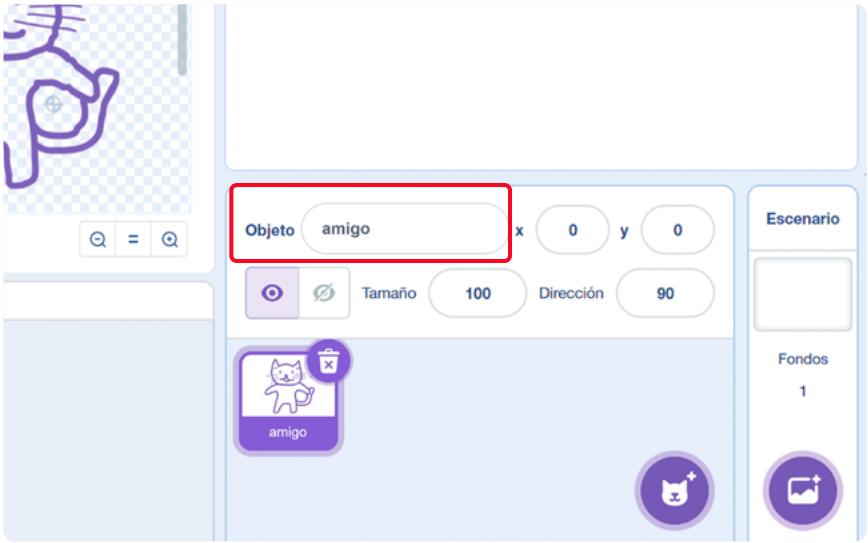
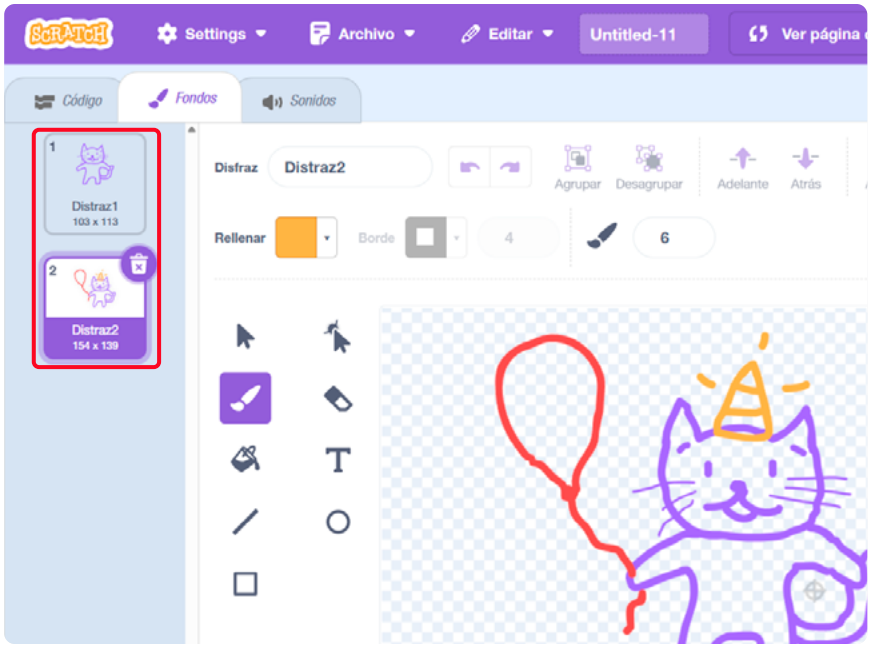




Figura 13. Objeto con un cambio de disfraz



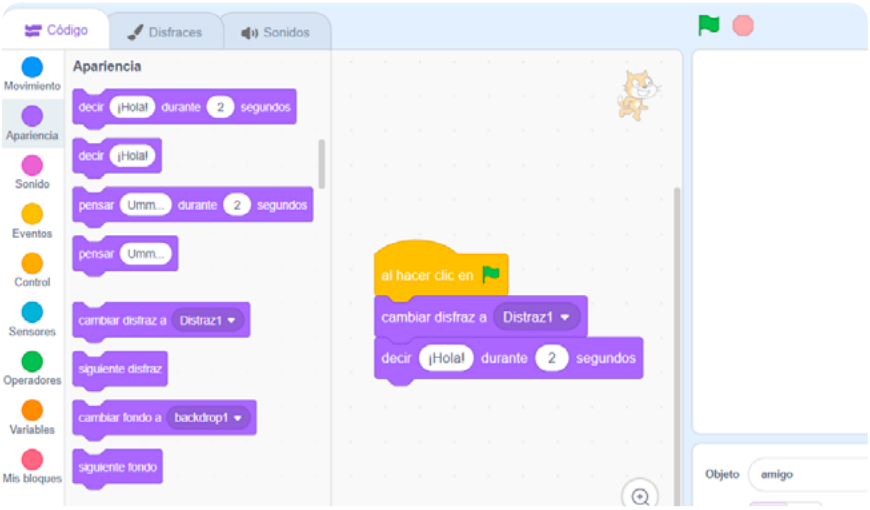
5. Una vez hayan dibujado libremente a Amigo, hagan un disfraz para él. Dupliquen el disfraz1, haciendo clic derecho sobre este. Como resultado, se creará una copia llamada "disfraz2". Modifiquen la figura para mostrar un cambio en el objeto, ver *Figura 13*. Por ejemplo, el disfraz2 en la *Figura 13* es el mismo amigo diseñado previamente, pero cargando un globo y llevando un gorro de fiesta.

6. Seleccionen de nuevo disfraz1 y ahora hagan clic en la pestaña **Código** y seleccionen **Eventos.** Luego, hagan clic en el bloque **al hacer clic en la bandera verde** o **al presionar la bandera verde.**

7. Para asegurarse de que su objeto siempre empiece el programa con el disfraz1, agreguen un bloque **cambiar disfraz a disfraz1**. Este bloque lo encuentran en la categoría de **Apariencia**.

8. Desde esta misma categoría, arrastren el bloque **decir ¡Hola! durante 2 segundos**. Hasta ahora su programa debería verse como en la *Figura 14*.

Figura 14. Avance del programa hasta el paso 8



9 Continúen agregando bloques. Seleccionen de la categoría **Sensores** el bloque **preguntar ¿Cómo te llamas? y esperar**.

10 Seleccionen de los bloques de **control** el que se llama **esperar un segundo**.

Figura 15. Avance del programa hasta el paso 10



Figura 17. Pastel de cumpleaños de *Scratch*



11 ¡Denle más animación al amigo que dibujaron! Vayan a la pestaña **Código** y seleccionen la categoría de bloques de **Apariencia**. Desde allí seleccionen el bloque **pensar Umm... durante 2 segundos**.

Figura 16. Código hasta el punto 11



Ahora comprueben que su programa funciona y hace lo que se espera. Den clic en la bandera verde ubicada en la parte superior derecha.

Recuerden lo que debería hacer su programa. Piensen luego en estas preguntas:

> *¿Qué quieren que haga el objeto ahora?*
> *Lean la situación descrita y discutan: ¿Pueden predecir los siguientes bloques?*

12 Llegó el momento de recordar lo que hiciste la clase pasada. Queremos que nuestro objeto se ubique en otro lugar después de realizar las acciones anteriores. Entonces necesitamos agregar un bloque que lo mueva hacia dónde queremos.

13 Agreguen un nuevo objeto, el pastel de cumpleaños de la *Figura 17*, y añadan el código para llevar a su Amigo hasta el pastel. Si no recuerdan cómo hacerlo, pueden revisar la sesión anterior.

Cuando tengan lista esta parte, prueben nuevamente haciendo clic en la banderita verde.

14 Finalmente, hagan que Amigo les lleve a una fiesta cuando toque el pastel. Para eso deben programar a Amigo para que cambie de disfraz si toca el pastel.

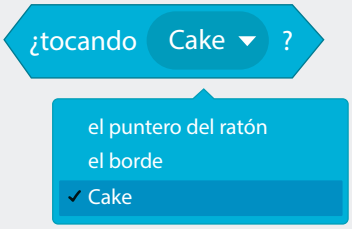
Para ello, seleccionen la categoría **Control**, y usen un bloque condicional.

Figura 18. Bloque de control si... entonces



15 Luego, vayan a **Sensores**, y elijan el bloque **tocando** y seleccionen **Cake**.

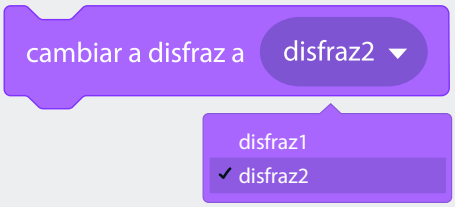
Figura 19. Bloque de control ¿tocando Cake?



Ubiquen este bloque en el espacio hexagonal del bloque **si... entonces.**

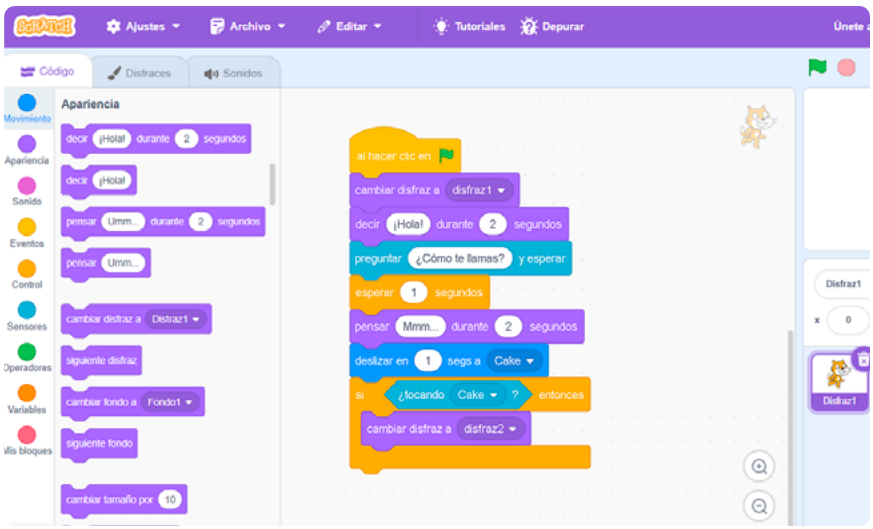
16 Vayan ahora a la categoría **Apariencia**, seleccionen “disfraz2” en el siguiente bloque y arrástrenlo al espacio del bloque **si... entonces.**

Figura 20. Bloque cambiar disfraz



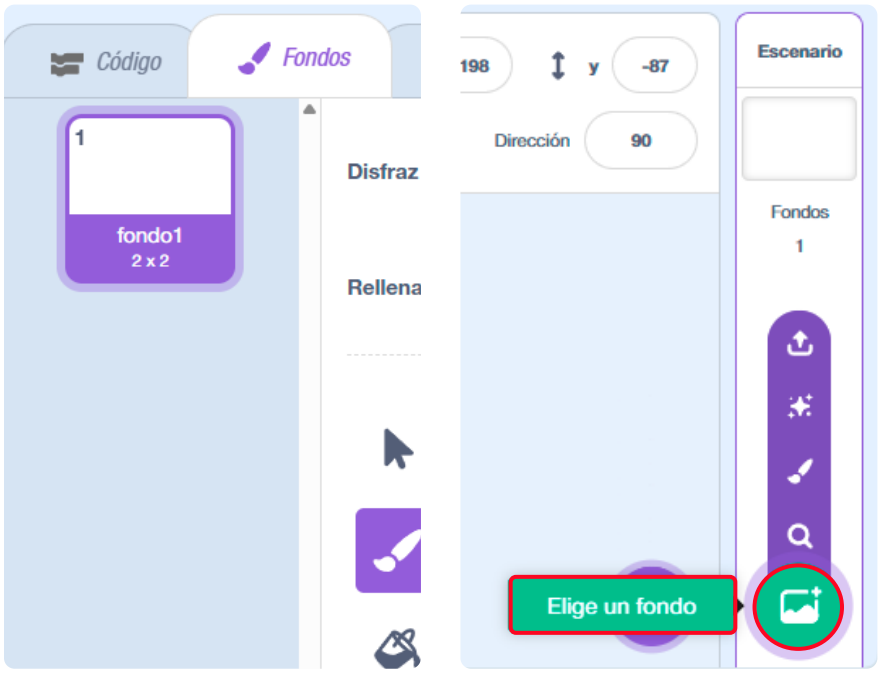
Su programa para el personaje debería verse como en la *Figura 21.*

Figura 21. Avance del programa hasta el punto 16



17 Finalmente, elijan un fondo y prográmenlo. Para esto deben hacer clic en **Escenario,** en la esquina inferior derecha del editor.

Figura 22. Selección de fondo



Luego, hagan clic en la pestaña “Fondos” y elijan un fondo. Por ejemplo, busquen el fondo llamado “Party”.

Figura 23. Selección del fondo Party

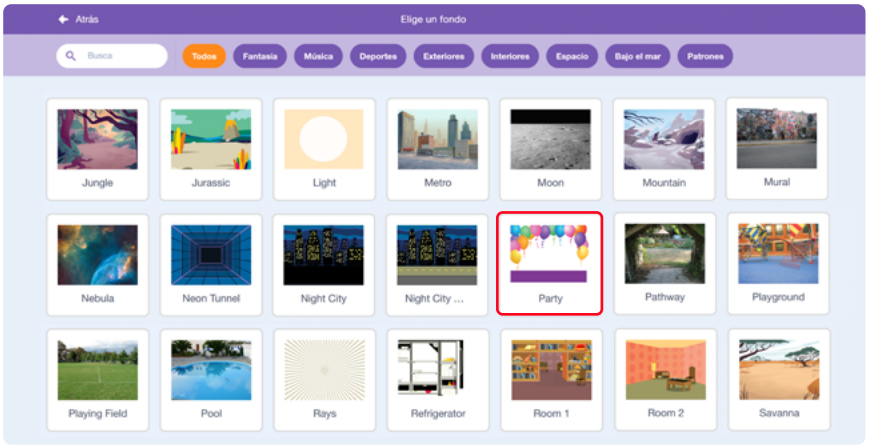
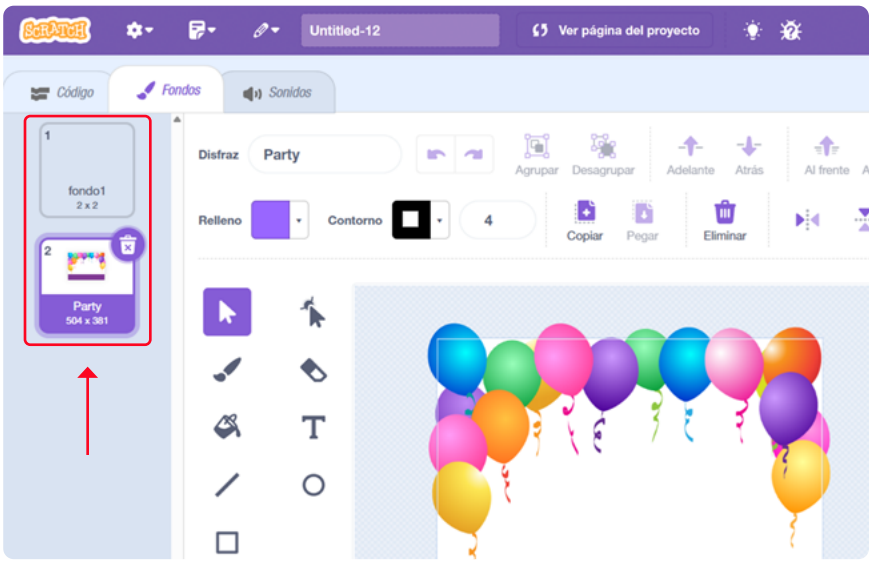
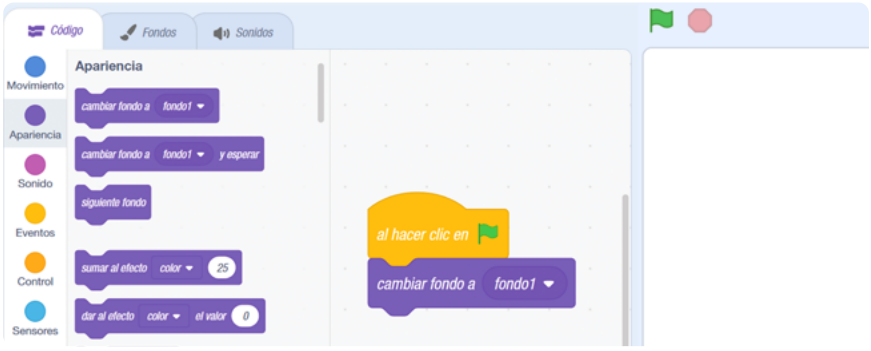


Figura 24. Fondo 1 en blanco y fondo 2 Party



18 Ahora, vayan nuevamente a la pestaña **Código** y armen el siguiente bloque, que se aplicará al fondo seleccionado. Este permitirá que cada vez que reinicien el programa, este vuelva al fondo blanco inicial.

Figura 25. Código para reiniciar con el fondo blanco



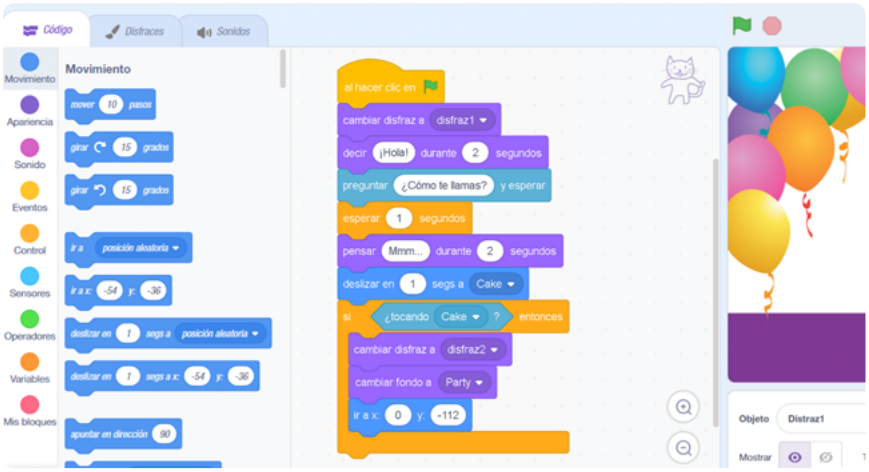
Party2



19 Seleccionen de nuevo a Amigo y en la categoría Apariencia, elijan el bloque **cambiar fondo a Party** dentro del **si... entonces**. Después, agreguen un bloque para que vuelva al centro del escenario.

Su programa completo debería verse como en la *Figura 26*.

Figura 26. Programa completo



¡Ahora prueben su programa! Y antes de terminar, hagan que Amigo diga “¡Feliz cumpleaños!”.

Glosario

Objeto: en programación, un objeto tiene un estado y un comportamiento. Pueden ser los personajes o elementos a los que se les modifica su comportamiento a partir de la programación.

Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

De forma individual, regresa a revisar los aprendizajes esperados. Elige la opción de respuesta que mejor describa lo que alcanzaste.

1. Puedes identificar y modificar objetos y sensores en *Scratch*?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

2. ¿Puedes elaborar un programa que realice acciones complejas mediante el lenguaje de bloques *Scratch*?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

3. ¿Puedes predecir el resultado de programas sencillos?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

> **Si tus respuestas fueron "Parcialmente" o "Aún no", vuelve a las actividades propuestas en *Scratch*. Luego, discute con tu compañera y compañero de grupo lo que se hizo en cada momento de la actividad y el rol al que correspondía. Si todavía te quedan dudas, consúltale a tu docente.**

Ahora te proponemos un reto usando una rutina llamada pensar, presentar e integrar (P-P-I):

Primero respondemos individualmente. Luego, cada persona en su turno le presenta al resto del grupo sus respuestas. Finalmente, el grupo integra una respuesta unificada.

Las preguntas que te proponemos son las siguientes:

¿Te enfrentaste a dificultades en el desarrollo de tu programa?, ¿cuáles? ¿Cómo las resolviste?

¿Cómo se compara el programa final con lo que imaginaste cuando leíste la situación?

¿Qué mejoras le podrías hacer al programa?

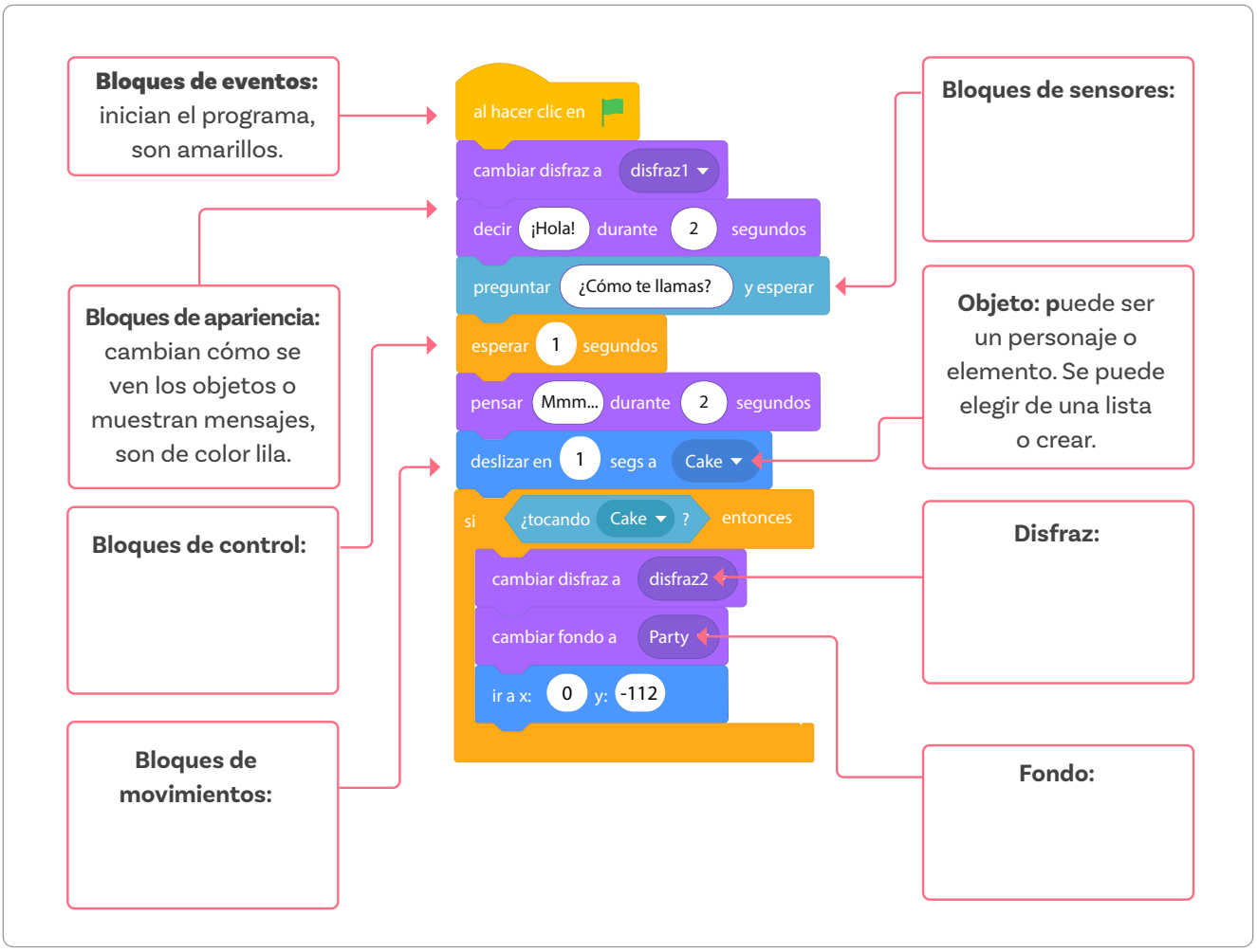
¿Qué le sugerirías a Saray para hacer la animación aún más especial?

Si regresas, al reto inicial que se propone en el *Anexo 1.2*, ¿para qué crees que te servirá lo aprendido en esta sesión?

Aprovecha este espacio final para hacerte un esquema en el que resumas lo que aprendiste, por ejemplo, colocando definiciones de palabras y algún ejemplo de uso de los bloques usados. En la siguiente página se presenta un ejemplo usando el código desarrollado en esta sesión. Lo puedes completar o puedes hacer algo parecido con tu propio código.

Comparte con tus compañeros y compañeras las respuestas y conversen con su docente.

Gráfico de anclaje. Sesión 2, guía 2, grado 6°





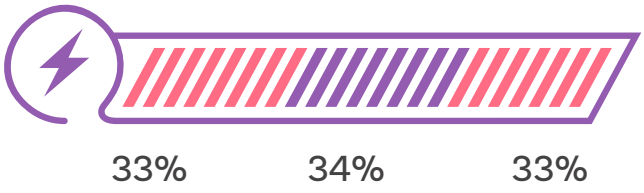
Sesión 3

Aprendizajes esperados

Con las actividades de esta sesión se espera que puedas avanzar en:

- Entender qué son las funciones y cómo se utilizan para agrupar y reutilizar código.
- Definir funciones.
- Combinar funciones con la extensión de lápiz para crear patrones y dibujos.

Duración sugerida



33% 34% 33%



Material para la sesión

Material por grupo:

- ○ Dispositivo con acceso a *Scratch*.

Lo que sabemos, lo que debemos saber



Esta sección corresponde al 33% de avance de la sesión

Ya has aprendido a reconocer que los programas tienen un inicio, eventos, objetivos y resultados esperados; la importancia de la secuencia de instrucciones, y a usar bloques de movimiento y eventos para crear programas sencillos usando *Scratch*. Pero, para programas más complejos, donde hay muchos movimientos u objetos, es difícil y agotador tener que repetir tantas veces la misma serie de bloques.

Toma un lápiz y para dibujar un cuadrado. Veamos cuántos pasos o acciones tenemos que realizar para dibujar un cuadrado de dos centímetros, por ejemplo.

1. Bajar el lápiz.
2. Mover el lápiz en línea recta dos centímetros.
3. Girar 90 grados en contra de las manecillas del reloj.
4. Mover el lápiz en línea recta dos centímetros.
5. Girar 90 grados en contra de las manecillas del reloj.
6. Mover el lápiz en línea recta dos centímetros.
7. Girar 90 grados en contra de las manecillas del reloj.
8. Mover el lápiz en línea recta dos centímetros.
9. Subir el lápiz.

Si dibujamos 5 cuadrados, entonces serían 45 pasos, lo cual puede ser demasiado largo y difícil. Lo mismo ocurriría si tuviéramos que hacer una simulación programada con bloques de *Scratch*.

Figura 1. Ícono de bloques de funciones en *Scratch*



Para esto existen las funciones. Las funciones son bloques de código que agrupan una serie de instrucciones que pueden ser reutilizadas en diferentes partes del programa. En *Scratch*, puedes encontrarlas como **Mis bloques,** ver *Figura 1.*

Utilizar funciones nos permite evitar la repetición de código y hacer programas más eficientes. En esta sesión veremos un ejemplo sencillo de cómo utilizar funciones para realizar tareas repetitivas, como dibujar varios cuadrados en *Scratch.*

Enlace

Acceso a *Scratch*

Glosario

Funciones: son bloques de código personalizados que agrupan una serie de instrucciones para ser reutilizadas en diferentes partes del programa.

Manos a la obra Conectadas

Esta sección corresponde al 67% de avance de la sesión

Organízate en pareja con otra compañera o compañero, siguiendo las indicaciones de tu docente.

Vayan a scratch.mit.edu y hagan clic en **Crear** para iniciar el nuevo proyecto.

Para este proyecto van a añadir la extensión **Lápiz**, que pueden ver en la *Figura 2.* Con esa extensión se le permite a un objeto, como gato, dibujar, agregando una categoría de comandos con ese fin.

Figura 2. Extensión Lápiz en *Scratch*

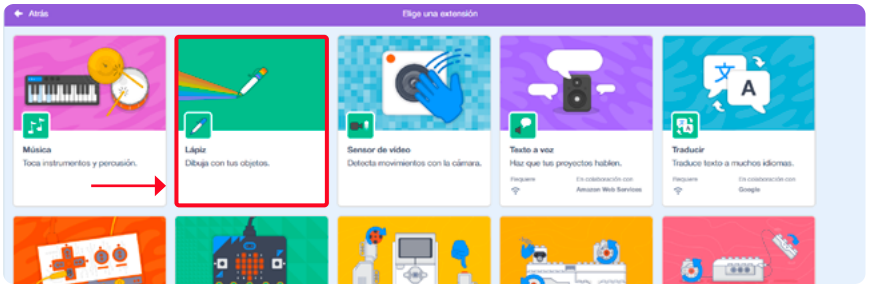
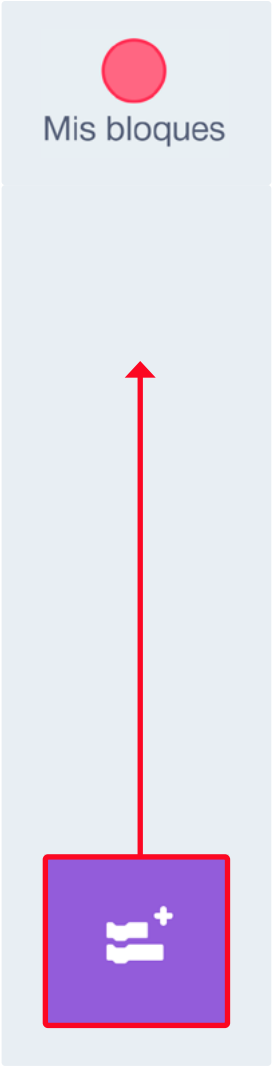


Figura 3. Ícono añadir extensión en *Scratch*



Deben agregarla haciendo clic en el ícono **Añadir extensión** en la parte inferior izquierda de la pantalla, que se muestra en la *Figura 3*.

En primer lugar, van a crear la función para dibujar los cuadrados.

1. Tengan en cuenta los primeros pasos para dibujar el cuadrado. Utilicen los bloques **bajar lápiz, mover 10 pasos, girar a la derecha 15 grados**. Modifiquen el código para variar el número de pasos. Sin embargo, asegúrense de que el giro a la derecha sea de 90 grados. Si dejan el movimiento en 40 pasos, su programa debería verse exactamente como el de la *Figura 4*.

Figura 4. Avance parcial del código para crear los cuadrados



2. Así como otros lenguajes de programación, *Scratch* cuenta con un bloque de control llamado repetir. Con este bloque pueden repetir las acciones las veces que necesites. En este caso, como el cuadrado tiene 4 lados, las instrucciones deben repetirse 4 veces. Para eso, añadan el bloque **repetir 10** que encuentran entre los bloques de control.

   Añadan el último bloque: **subir lápiz**.

3. Su programa debería verse como el de la *Figura 5*. Tengan en cuenta que pueden cambiar la cantidad de pasos que avanza el lápiz, dependiendo de qué tan grande quieran el cuadrado.

Figura 6. Botón Crear un bloque, que permite crear una nueva función

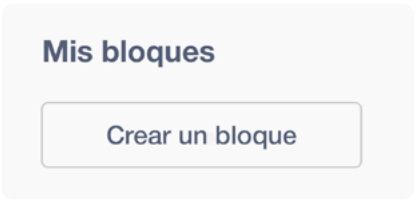
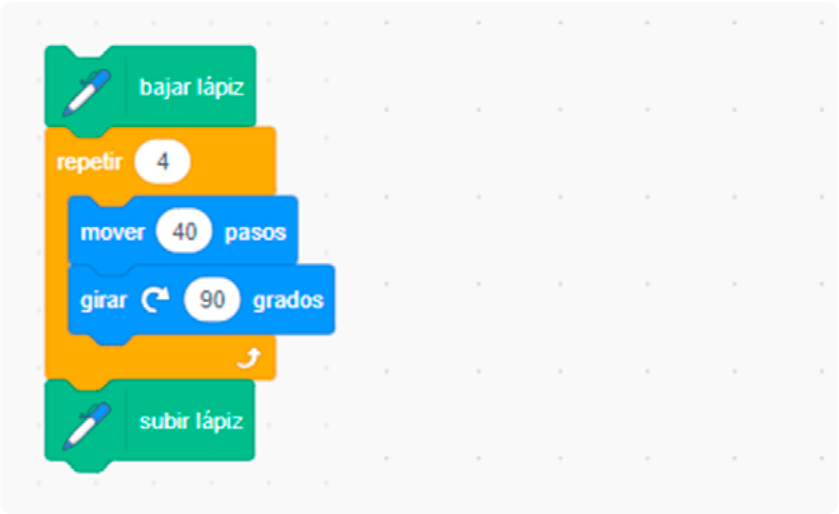


Figura 5. Avance del código para dibujar un cuadrado con *Scratch*



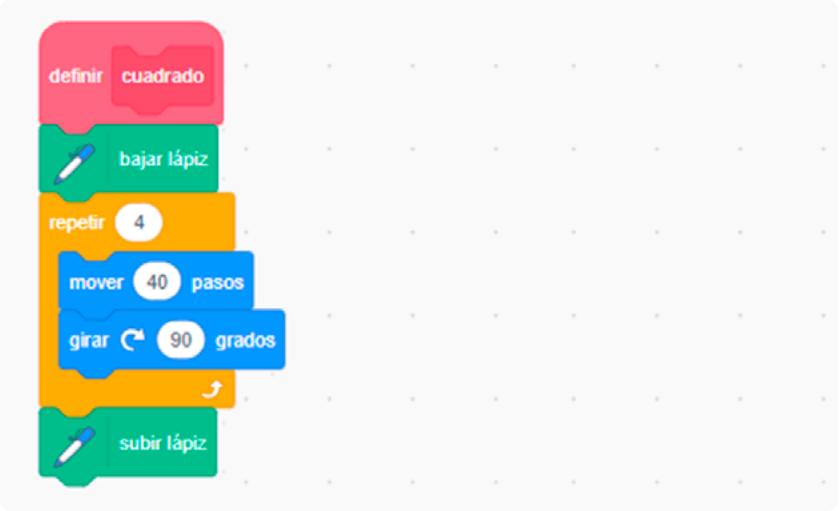
4 Hagan clic en **Mis bloques** y creen un bloque que se llame "cuadrado", ver *Figuras 6* y *7*.

Figura 7. Creación de la función cuadrado



5 Añadan el bloque **definir cuadrado** al inicio de la secuencia que acaban de hacer. Este bloque es la función que utilizarán para dibujar los cuadrados en este programa. Su código debería verse como en la *Figura 8*.

Figura 8. Código de la función cuadrado

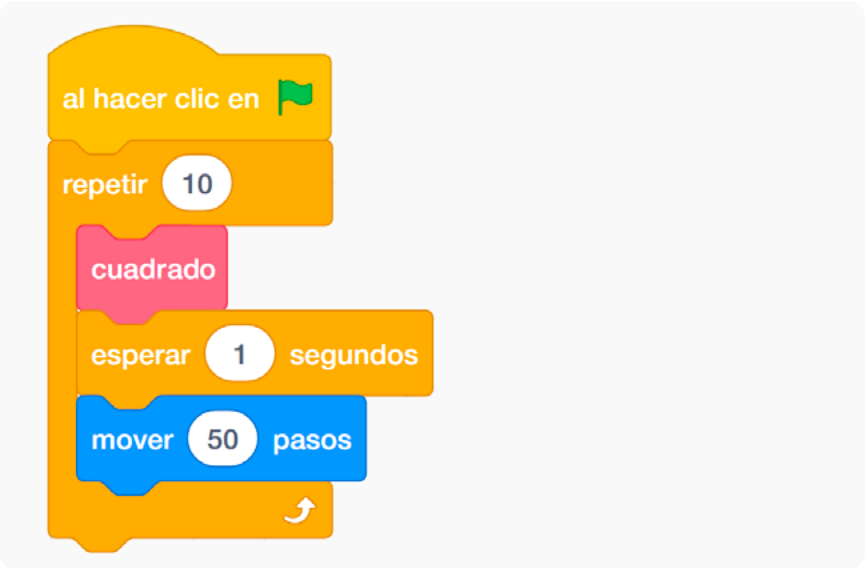


Ahora, hagan la secuencia para dibujar varios cuadrados usando esta nueva función.

1. Seleccionen el bloque de control **al hacer clic en bandera verde.**
2. Añadan el bloque **repetir 10.**
3. Dentro del bloque repetir agreguen el bloque **cuadrado**, que se encuentra en **Mis bloques**.
4. Para que vaya pausado, dentro del bloque repetir añadan el bloque **esperar 1 segundos**.
5. Para separar los cuadrados que el gato va a dibujar, dentro del bloque repetir añadan el bloque **mover 50 pasos**.

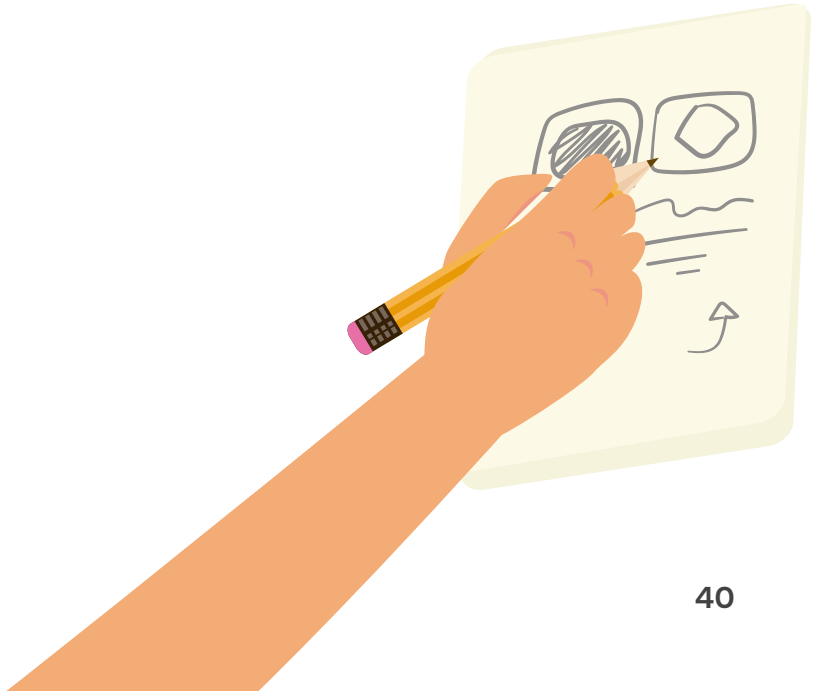
Ya tienen listo un programa construido con su propia función. Debería verse como el de la *Figura 9*.

Figura 9. Código para dibujar 10 cuadrados, dejando una separación de 50 pasos entre ellos



Pueden modificar a su gusto las variables que se encuentran en esta secuencia y ver cómo cambia el resultado. Intenten, por ejemplo, eliminar el bloque “esperar” o añadir un nuevo bloque de “girar 15 grados”.

Si cuentan con el tiempo suficiente, integren a su programa lo que han aprendido hasta el momento. Por ejemplo, pueden hacer que un personaje camine hasta tocar un lápiz y que solo cuando lo toque el lápiz dibuje 10 cuadrados.



Antes de irnos



Esta sección corresponde al 100% de avance de la sesión

De forma individual, regresa a revisar los aprendizajes esperados. Elige la opción de respuesta que mejor describa lo que alcanzaste.

1. ¿Puedes entender qué son las funciones y cómo se utilizan para agrupar y reutilizar código?

- ◯ Sí
- ◯ Parcialmente
- ◯ Aún no

2. ¿Puedes definir funciones?

- ◯ Sí
- ◯ Parcialmente
- ◯ Aún no

3. ¿Puedes combinar funciones con la extensión de lápiz para crear patrones y dibujos?

- ◯ Sí
- ◯ Parcialmente
- ◯ Aún no

Si tus respuestas fueron "Parcialmente" o "Aún no", vuelve a las actividades propuestas en *Scratch*. Luego, discute con tu compañera y compañero de grupo lo que se hizo en cada momento de la actividad de creación de cuadrados y el resultado que el programa iba generando. Si todavía te quedan dudas, consúltale a tu docente.

Ahora completa las siguientes frases y luego espera indicaciones de tu docente para compartir tus respuestas con el resto de la clase.

Si mi código no usara las funciones, _____

_____

_____

Si tuviera que modificar el código para hacer cuadrados más pequeños solo tendría que _____

_____

_____

Si quisiera dibujar rectángulos, _____

_____

_____

Lo que aprendí sobre funciones me ayudará a resolver el reto del *Anexo 1.2* porque _____

_____

_____

Sigue las indicaciones para discutir las respuestas que tú y tus compañeros y compañeras dieron a las preguntas anteriores.

Aprovecha este espacio final para hacerte un esquema en el que resumas lo que aprendiste, por ejemplo, colocando definiciones de palabras y algún ejemplo de uso de los bloques usados.

Comparte con tus compañeros las respuestas y conversen con su docente.

Sesión 4

Aprendizajes esperados

Con las actividades de esta sesión se espera que puedas avanzar en:

- Elaborar un programa con *Scratch* para simular fenómenos complejos.
- Identificar códigos y bloques necesarios para elaborar el programa.

Duración sugerida



15% 70% 15%



Material para la sesión

Material por grupo:

- ○ Anexo 4.1.
- ○ Dispositivo con acceso a *Scratch*.

Enlace

Acceso a *Scratch*

Anexo

Anexo 1.2



Lo que sabemos, lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

Ya has aprendido a reconocer que los programas tienen un inicio, eventos, objetivos y resultados esperados, la importancia de la secuencia de instrucciones y la detección de eventos, y a usar bloques de movimiento y eventos para crear programas sencillos usando *Scratch*. En esta sesión iniciarás la elaboración de un modelo en *Scratch* para simular cómo se propaga un virus.

Al usar simulaciones para entender cómo funcionan las cosas, podemos crear nuevas ideas o mejorar las que ya tenemos. También podemos hacer "experimentos imaginarios" para ver si nuestras ideas son correctas, sin tener que hacer experimentos reales.

Además, de ser útiles para aprender, programar una simulación, aunque sea simple y no tan realista, es una excelente manera de practicar nuestras habilidades de programación y creatividad.

Ahora con todos los aprendizajes alcanzados hasta aquí, se propone el siguiente reto para ti: vas a elaborar una simulación para estudiar cómo un virus como los que ocasionan la gripe se propaga.

Para comprender cómo el programa elaborado con *Scratch* puede ayudar a estudiar la propagación del virus realizarás un programa utilizando todo lo que has aprendido hasta ahora.

Manos a la obra Conectadas

Esta sección corresponde al 85% de avance de la sesión

Organízate en pareja con otra compañera o compañero, siguiendo las indicaciones de tu docente.

Vuelvan a la definición del reto del *Anexo 1.2*. Léanlo detenidamente e identifiquen qué partes del programa ya son capaces de desarrollar.

Discutan cómo se imaginan el programa. Revisen sus notas de clases anteriores.

> *¿Pueden identificar cosas en común con las actividades que han desarrollado hasta el momento?*

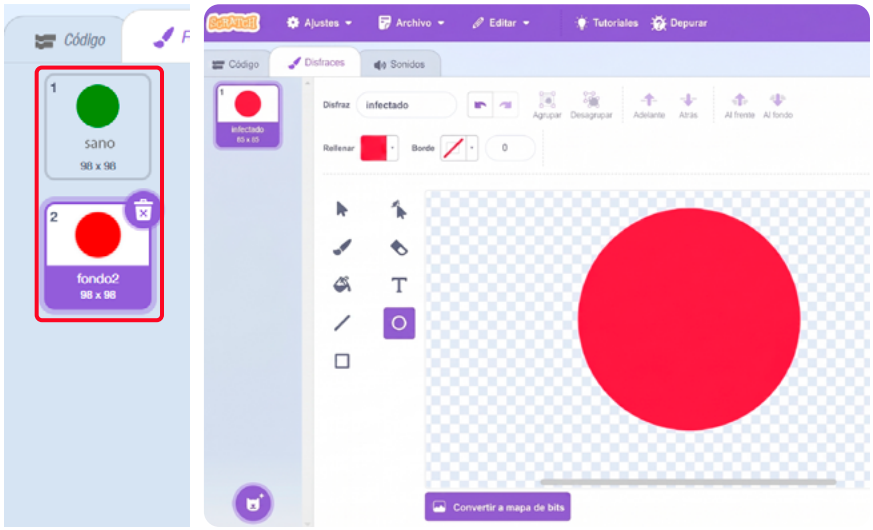
Una vez tengan claro lo que se espera lograr, pueden seguir las instrucciones para comenzar la simulación. Recuerden que tienen sus notas y los ejemplos pasados para apoyarse.

1. Vayan a scratch.mit.edu y hagan clic en **Crear** para iniciar el nuevo proyecto.

2. Para este proyecto tendrán que representar personas sanas y personas contagiadas. Para representarlas, usen como objeto un círculo y creen dos disfraces de colores diferentes, por ejemplo, uno verde para representar las personas sanas y uno rojo para representar las infectadas, ver *Figura 1*. Recuerden lo visto en la sesión 2 de esta guía.



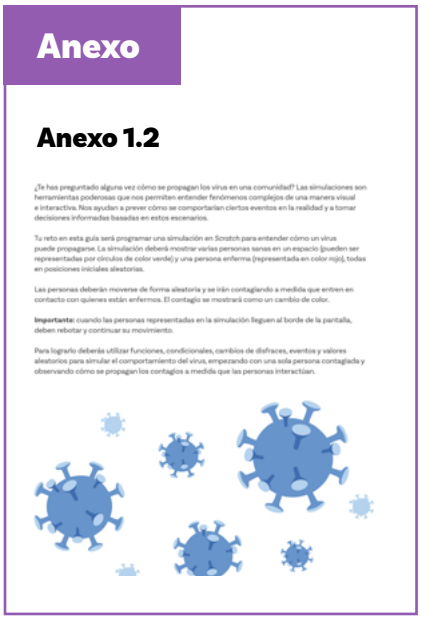
Figura 1. Disfraces del objeto creado



3. Recuerden nombrar cada círculo según lo que representan antes de salir de esta pestaña. Debería verse como en la *Figura 1*.

Anexo

Anexo 1.2

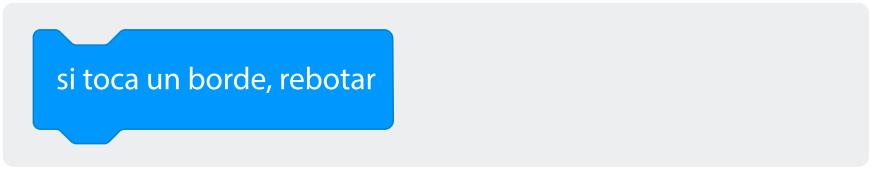


Ahora, seleccionen el disfraz "infectado" y programen el movimiento básico de este círculo para que se mueva y rebote por la pantalla.

4. Empiecen seleccionando el bloque adecuado en la paleta de **Eventos** para que el programa se ejecute al presionar la bandera verde.

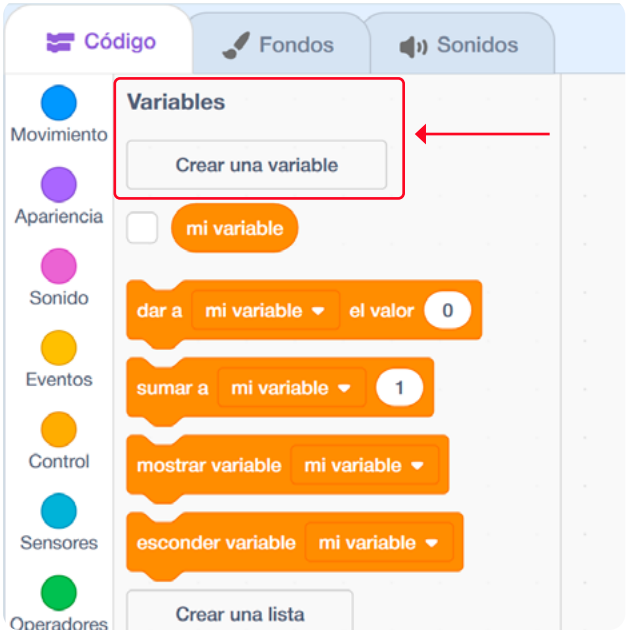
5. Añadan un bloque para crear un bucle **por siempre** que haga que el círculo se mueva y rebote cada vez que toque el borde de la pantalla. Usen el bloque que aparece en la *Figura 2* para el rebote.

Figura 2. Bloque de movimiento si toca un borde, rebotar



6. Para ajustar la velocidad en la que se mueve el círculo que representa la persona infectada, pueden crear una variable llamada "velocidad", ver *Figuras 3 y 4*.

Figura 3. Bloque Variables



Si lo desean pueden usar colores diferentes para representar a las personas sanas y las infectadas en su simulación. Lo importante es que etiqueten claramente los disfraces que usarán en cada caso.

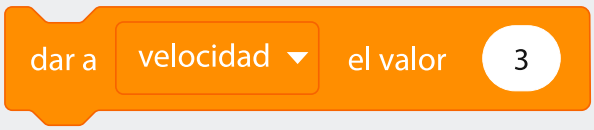


Figura 4. Creación de la variable velocidad



7 Añadan el siguiente bloque antes del bucle **por siempre**. Pueden probar qué tan rápido se mueve el objeto dándole inicio al programa y ajustando el valor a su preferencia. En la figura 5 se le dio a la variable velocidad el valor 3.

Figura 5. Bloque de Variables, **dar a velocidad el valor 3**



8 En el bloque **mover _ pasos**, cambien el número por la variable velocidad. Ahora, su programa debería verse como en la *Figura 6*.

Figura 6. Código para que el objeto infectado rebote por la pantalla, moviéndose al valor asignado a la variable velocidad



Figura 7. Bloques de movimiento **ir a posición aleatoria** y **apuntar en dirección 90**

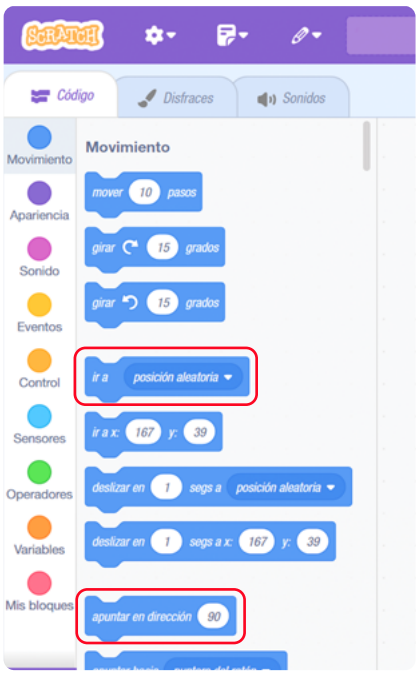
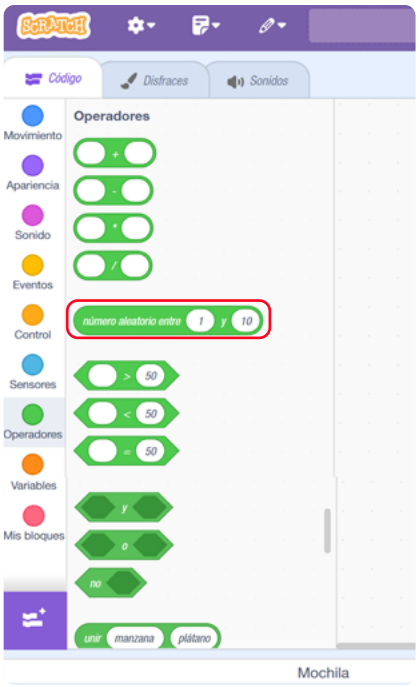


Figura 8. Bloque elegir número al azar entre 1 y 10

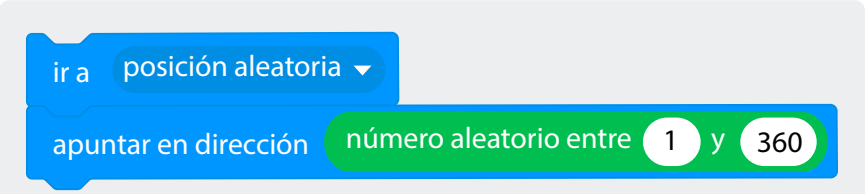


9. Para hacer que el círculo aparezca en una posición aleatoria y se mueva también de forma aleatoria, como un virus, inserten los bloques de movimiento **ir a posición aleatoria** y **apuntar en dirección 90** antes del bucle **por siempre**, ver *Figura 7*.

10. Cambien el número 90 del bloque por el bloque **elegir número al azar entre __ y __** que encuentran en la categoría de **Operadores**, ver *Figura 8*.

11. Cambien los números de este bloque por los números 1 y 360 respectivamente, como se muestra en la *Figura 9*. Este valor representa los grados del ángulo de giro a los que podrá dirigirse el objeto que representa a la persona infectada.

Figura 9. Código parcial para que el objeto infectado apunte de forma aleatoria variando el ángulo de movimiento entre 1 y 360 grados



12. Ahora, cada vez que ejecuten el programa, el círculo se moverá a una nueva ubicación y apuntará en una dirección aleatoria.

13. Su programa debe verse como en la *Figura 10*.

Figura 10. Código creado hasta este punto



Discutan como grupo:

¿? *¿Qué hace este programa?*

Este código hace que el objeto que representa a una persona infectada se mueva aleatoriamente, al azar, por el escenario. ¡Pruébenlo!

Este es un código para un solo objeto, pero para crear la simulación de un contagio van a necesitar más de un objeto infectado. En la próxima sesión verán cómo hacerlo. Por esto, es importante que guarden el archivo de este proyecto en el dispositivo. Para esto deben ir a la parte superior de la ventana, dar clic en **Archivo** y luego en **Guardar en tu ordenador**.





Figura 11. Pasos para guardar el programa en el computador



Anexo

Anexo 4.1



3 páginas

Antes de irnos

Esta sección corresponde al 100% de avance de la sesión

De forma individual, regresa a revisar los aprendizajes esperados. Elige la opción de respuesta que mejor describa lo que alcanzaste.

1. ¿Puedes elaborar un programa con *Scratch* para simular fenómenos complejos?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

2. ¿Puedes identificar códigos y bloques necesarios para elaborar el programa?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

> **Si tus respuestas fueron "Parcialmente" o "Aún no", vuelve a las actividades propuestas en *Scratch*. Luego, discute con tu compañera o compañero de grupo lo que significan los círculos que agregaron a la simulación y lo que hacen los bloques nuevos que usaron para crear el programa. Si todavía te quedan dudas, consúltale a tu docente.**

Trabaja en grupos de tres personas, según las indicaciones de tu docente.

Vayan al *Anexo 4.1* y completen la información con los bloques que hayan aprendido a utilizar o recordado al realizar las actividades de esta sesión. Traten de agregar, al menos, un bloque que corresponda a cada una de las categorías.

Su docente tomará luego algunos de estos bloques para mostrarlos a toda la clase, a modo de juego de memoria, a fin de verificar qué tanto recuerdan sobre el uso de estos bloques dentro de un programa.

Sesión 5

Aprendizajes esperados

Con las actividades de esta sesión se espera que puedas avanzar en:

- Elaborar un programa con *Scratch* para simular fenómenos complejos con varios agentes u objetos.
- Identificar códigos y bloques necesarios para elaborar un programa con varias funciones.

Duración sugerida



15% 70% 15%



Material para la sesión

Material por grupo:

- ○ Anexos 1.2 y 4.1.
- ○ Dispositivo con acceso a *Scratch*.

Lo que sabemos, lo que debemos saber

Esta sección corresponde al 15% de avance de la sesión

Ahora, con todos los aprendizajes alcanzados hasta aquí, se propone el siguiente reto para ti.

> *Terminar la simulación que se plantea en el Anexo 1.2, que sirve para estudiar cómo se propaga un virus como los que ocasionan la gripe u otras enfermedades.*



Lee de nuevo el reto que aparece en el anexo y también revisa lo que llevas del programa antes de continuar con la actividad.

Podrás comprender cómo el programa elaborado con *Scratch* aplica todo lo aprendido hasta el momento para completar el reto.

Manos a la obra Conectadas

Esta sección corresponde al 85% de avance de la sesión

Organízate en pareja con la misma persona y dispositivo con los que trabajaste la sesión anterior, a menos que tu docente te dé una instrucción diferente.



1. Vayan a scratch.mit.edu y hagan clic en Crear para iniciar el nuevo proyecto.

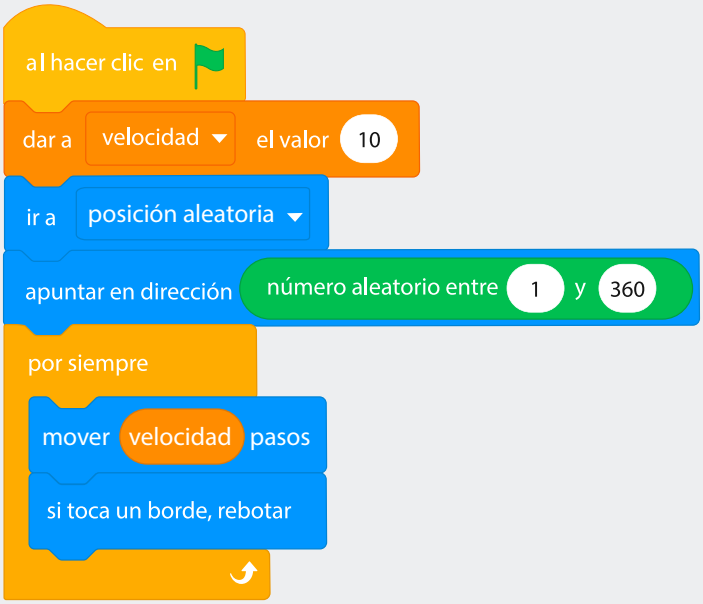
2. Hagan clic en Archivo y seleccionen Cargar desde tu ordenador. Abran el archivo que guardaron la sesión anterior. Seguirán utilizando este código para completar el reto.

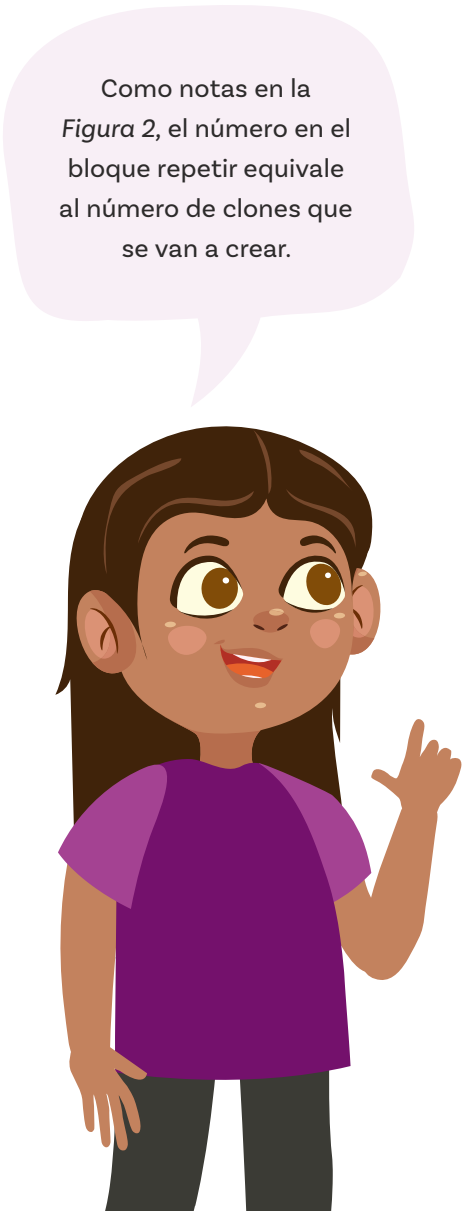
   Su programa debería verse como en la *Figura 1.*

Enlace

Acceso a *Scratch*

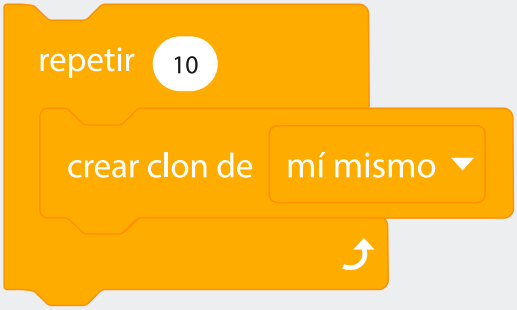
Figura 1. Código para simular el movimiento de una persona infectada





3 Para resolver el problema de los demás círculos que representan más pacientes infectados, vayan a la categoría **Control** e inserten el bloque **crear clon de mí mismo** dentro de un bloque **repetir 10**. Estos bloques deben quedar ubicados encima del bucle **por siempre** que tienen ya en su código. Esto permitirá que el objeto que representa a la persona contagiada se clone 10 veces.

Figura 2. Bloques para crear 10 clones del objeto que representa a las personas infectadas



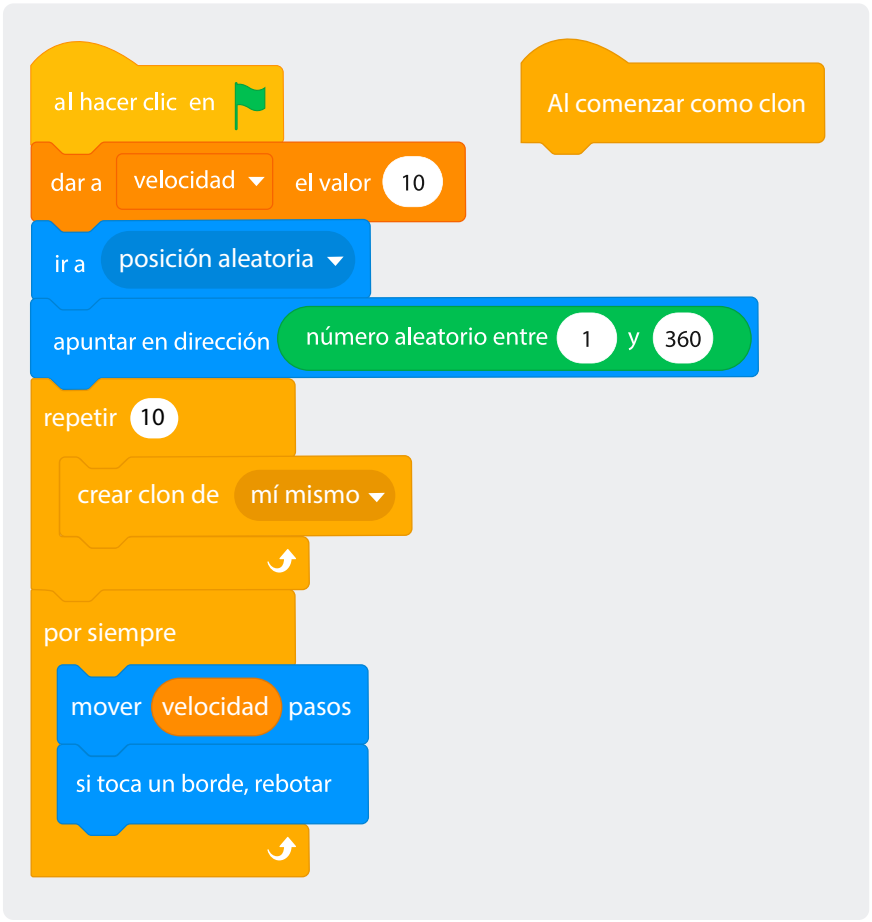
4 Empiecen una nueva secuencia de bloques, añadiendo el bloque de control **al comenzar como clon**, ver *Figura 3*.

Figura 3. Bloque de control al comenzar como clon.



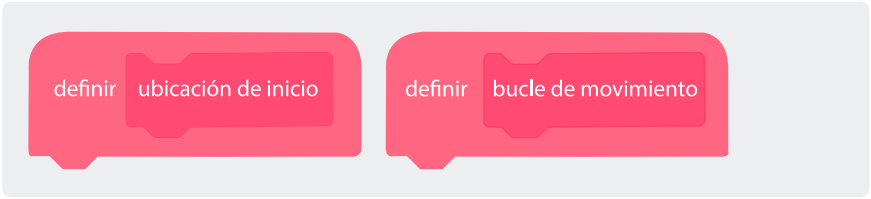
Este bloque inicia una nueva secuencia al lado del programa que ya tienen construido. Ahora su programa deberá verse como el de la *Figura 4*.

Figura 4. Avance del programa de simulación del virus, hasta el paso 4 de esta sección



5 Ahora, para hacer que la secuencia de programación se repita para todos los clones, deberán crear 2 funciones. Recuerden lo visto en la sesión 3 de esta guía. Llamen a la primera función “ubicación de inicio” y a la segunda “bucle de movimiento”. Cuando las hayan creado se deberían ver como las de la *Figura 5*.

Figura 5. Bloques de funciones “ubicación de inicio” y “bucle de movimiento”



6 Para definir lo que cada función debe hacer, tomen la secuencia de programación y muevan ciertas partes a estas, según lo que se indica en la *Figura 6*.

Figura 6. Redistribución del código inicial dentro de las dos funciones previamente creadas

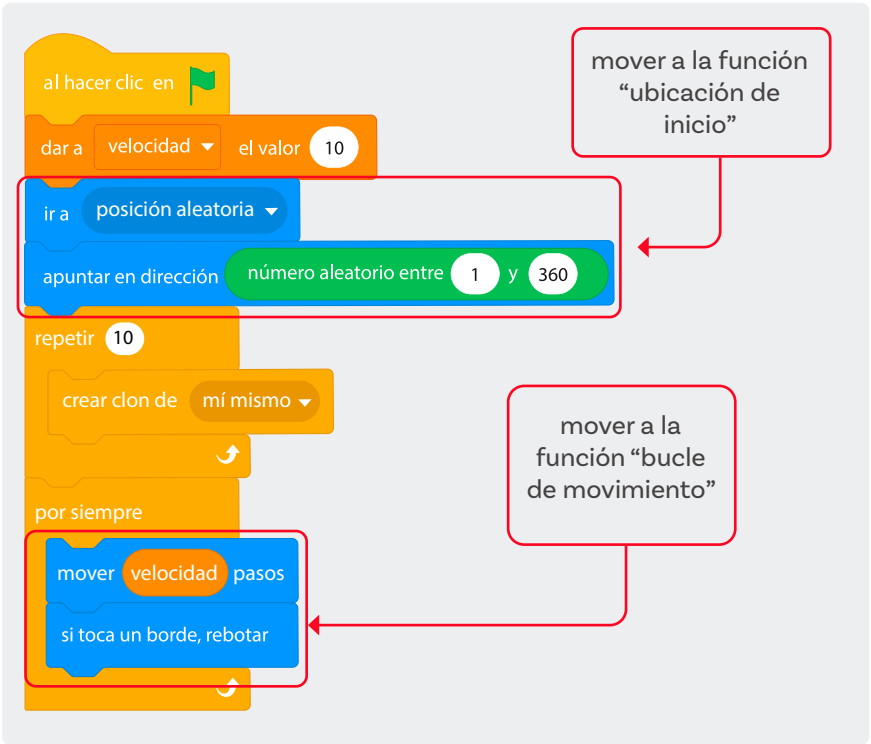
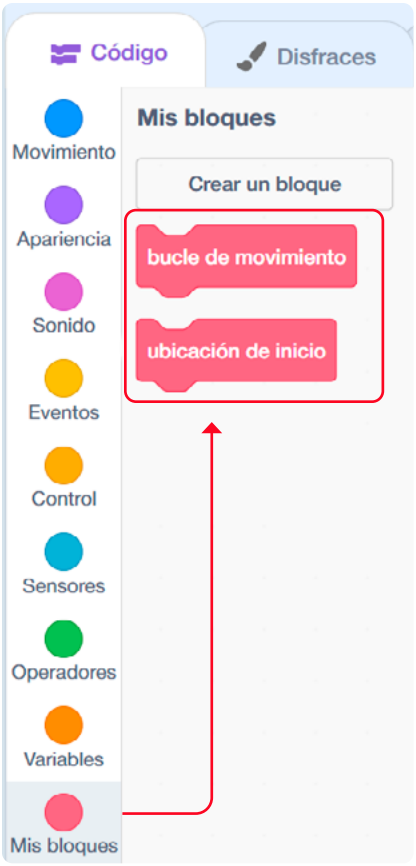


Figura 8. Ubicación de los bloques para llamar las funciones nuevas dentro del programa



Cuando hayan hecho los ajustes previos, las funciones en su programa deberían verse como las de la *Figura 7.*

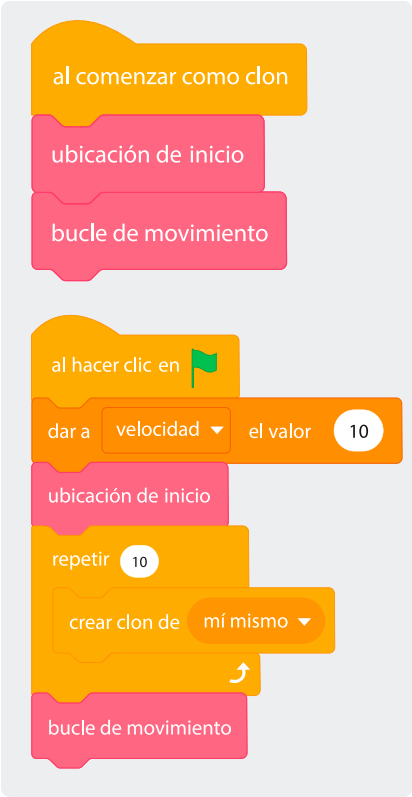
Figura 7. Funciones nuevas con su respectivo código



7. Ahora deben reemplazar las partes que movieron de las secuencias del inicio, añadiendo los nuevos bloques de funciones que encuentran en **Mis bloques**.

8. En el bloque **al comenzar como clon** hagan lo mismo. Su programa debería verse como el de la *Figura 9.*

Figura 9. Avance del programa hasta el punto 8: se reemplazó el código inicial por el llamado a las funciones creadas



9 Hagan clic en la bandera verde para ejecutar el programa. Podrán ver cómo hay varios círculos que representan personas infectadas moviéndose por toda la pantalla, pero todavía no se ve la simulación de las personas sanas.

Para esto deben seleccionar el bloque **cambiar disfraz a ___** en la categoría **Apariencia**. A la secuencia que determina la cantidad de clones, añadan el bloque **cambiar disfraz a infectado** y, a la que programa a los clones, **cambiar disfraz a sano**, ver *Figura 10*.

Figura 10. Código para representar a las personas sanas y la a infectada



Al hacer clic en la bandera podrán ver que solo hay una persona infectada y las demás están sanas. Solo falta simular el contagio de la persona infectada a las personas sanas.

10 Inserten el bloque de control **si __ entonces** en la función "definir bucle de movimiento".

¿?

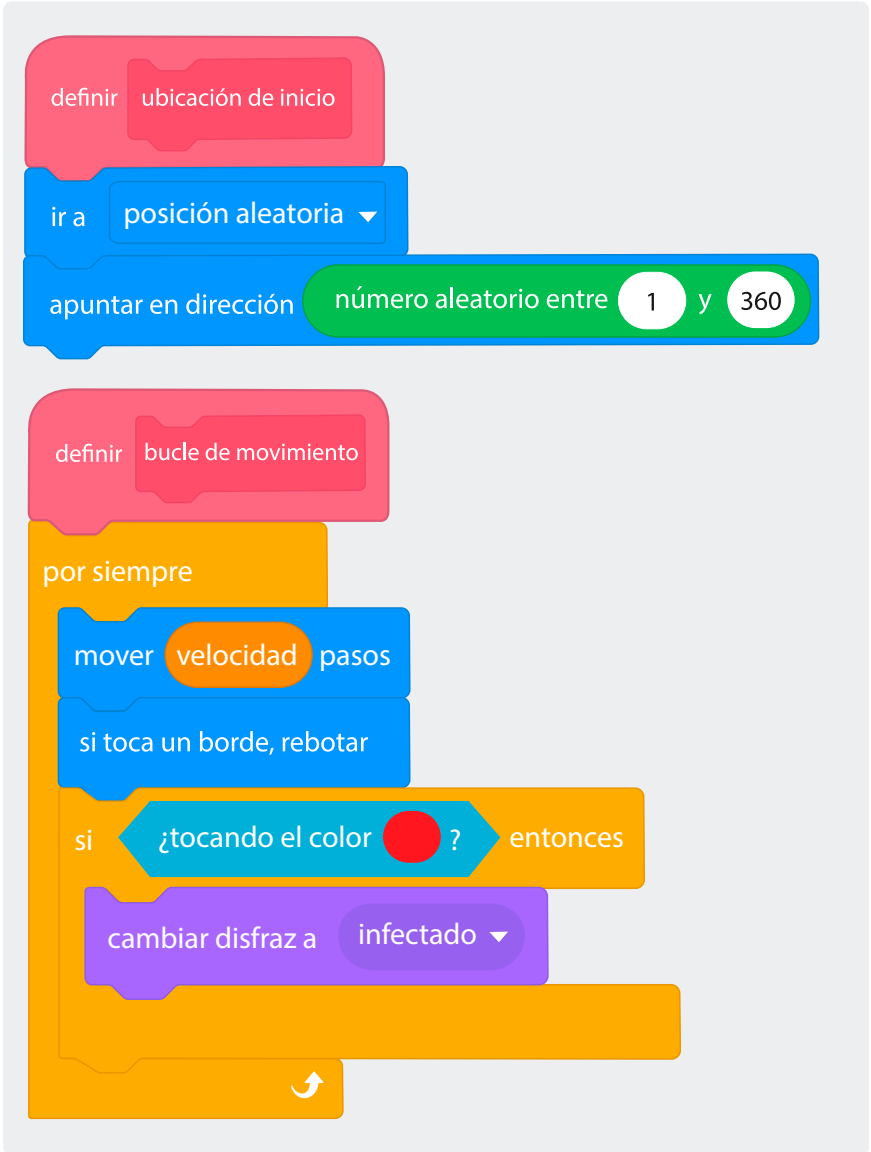
¿Qué bloques creen que deban ir como variables en el hexágono para que los círculos verdes que representan personas sanas cambien de color cuando se toquen con un círculo rojo (persona infectada)?

Recuerden lo visto en la sesión 2 sobre los sensores.

Como quizás lo descubrieron, la mejor opción es usar el bloque de sensor **¿tocando el color _?** y seleccionar el color rojo del círculo que representa a una persona contagiada. Lo otro que deben modificar es el bloque **cambiar disfraz a infectado**.

Una vez hagan estos cambios, verifiquen que su programa se vea como el de la *Figura 11.*

Figura 11. Ajustes finales de la función bucle de movimiento para el programa de simulación



Recuerden que pueden personalizar su simulación modificando las variables de velocidad de los objetos y cambiando la cantidad de clones que se generan.



Terminaron de programar la simulación del contagio de un virus. Pruébenla.

Computación y sociedad

En esta guía pudiste ver cómo se pueden visualizar fenómenos como la propagación de una enfermedad infecciosa.

La computación ayuda a mejorar la salud de las personas en muchas formas, dado que las personas que han aprendido computación escriben programas de ayuda en el sector médico.

Las y los profesionales de la salud deben saber cada vez más sobre computación.

Antes de irnos

Esta sección corresponde al 100% de avance de la sesión

De forma individual, regresa a revisar los aprendizajes esperados. Elige la opción de respuesta que mejor describa lo que alcanzaste.

1. ¿Puedes elaborar programa con *Scratch* para simular fenómenos complejos con varios agentes u objetos?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

2. ¿Puedes identificar códigos y bloques necesarios para elaborar el programa?

   - ◯ Sí
   - ◯ Parcialmente
   - ◯ Aún no

Anexo

Anexo 4.1



Si tus respuestas fueron "Parcialmente" o "Aún no", vuelve a las actividades propuestas en sobre la simulación en *Scratch*. Luego, discute con tu compañera o compañero de grupo lo que se hizo en cada momento de la actividad y lo que iba haciendo cada bloque. Si todavía te quedan dudas, consúltale a tu docente.

Para finalizar, piensa en cómo le explicarías a tu mamá, papá o alguna de las personas que te cuidan lo que hiciste en la clase. Haz un dibujo y agrega algunas palabras para que puedas explicar lo que son:

- ○ Los círculos rojos.
- ○ Los círculos verdes.
- ○ Lo que pasa cuando un círculo verde toca uno rojo.
- ○ El resultado después de un tiempo de ejecutar el programa de simulación.

Luego, escribe tus respuestas a las siguientes preguntas:

¿Cómo podrías modificar la simulación para cambiar los puntos por otros personajes?

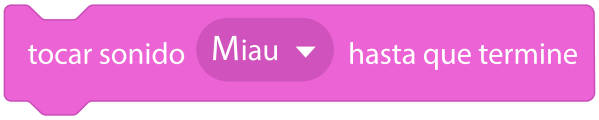
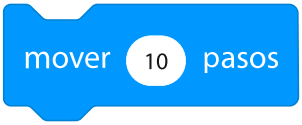
¿Qué podrías mejorarle al programa, a partir de lo que aprendiste en clases pasadas?

¿Por qué crees que se simulan los fenómenos y no se prueban en la vida real?

Sigue las instrucciones de tu docente para mostrarle tu dibujo al resto de la clase y para compartir tus respuestas.

Para finalizar, complementa el *Anexo 4.1* agregando los bloques y funciones nuevos que usaste en esta sesión.

Anexo 1.1. Tabla de acciones usadas en la sesión

Bloque de *Scratch*	Acción que realiza
	
	
	
	

Anexo 1.2 Reto

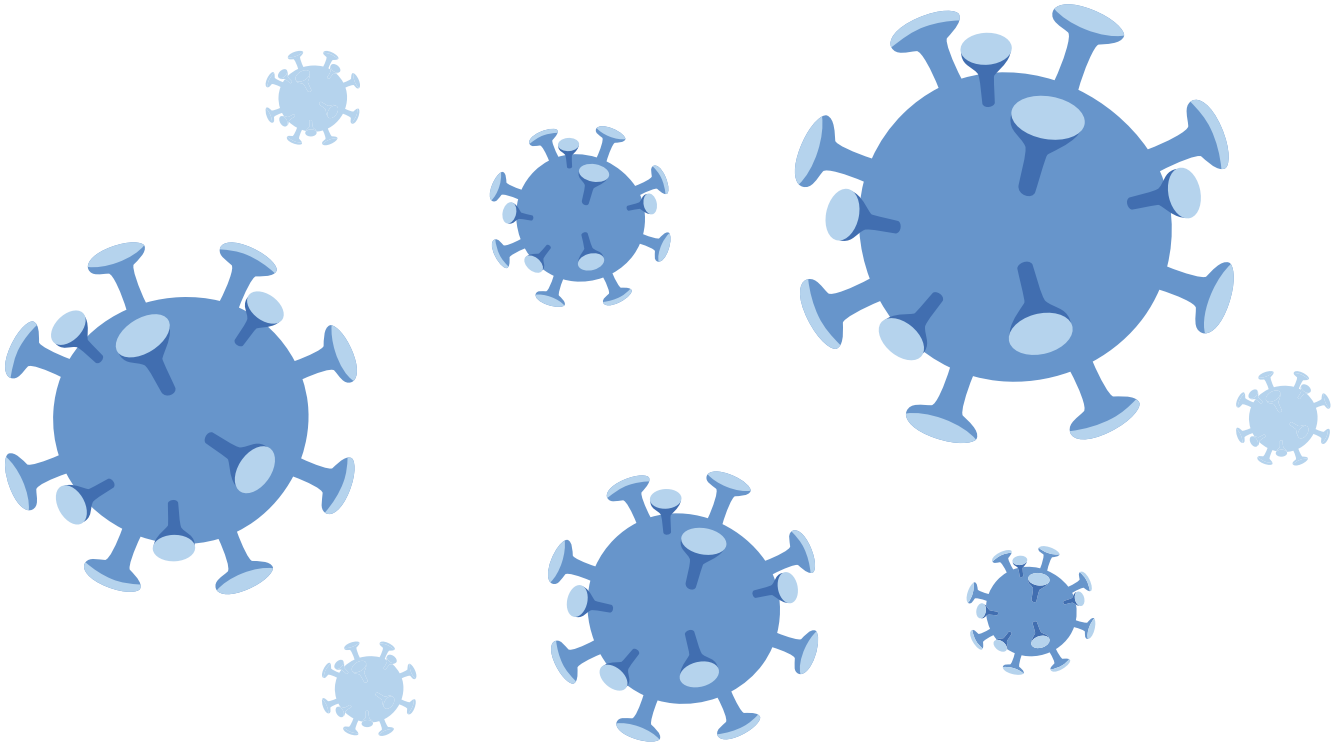
¿Te has preguntado alguna vez cómo se propagan los virus en una comunidad? Las simulaciones son herramientas poderosas que nos permiten entender fenómenos complejos de una manera visual e interactiva. Nos ayudan a prever cómo se comportarían ciertos eventos en la realidad y a tomar decisiones informadas basadas en estos escenarios.

Tu reto en esta guía será programar una simulación en *Scratch* para entender cómo un virus puede propagarse. La simulación deberá mostrar varias personas sanas en un espacio (pueden ser representadas por círculos de color verde) y una persona enferma (representada en color rojo), todas en posiciones iniciales aleatorias.

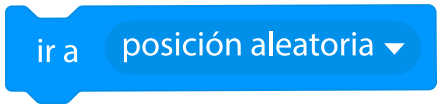
Las personas deberán moverse de forma aleatoria y se irán contagiando a medida que entren en contacto con quienes están enfermos. El contagio se mostrará como un cambio de color.

Importante: cuando las personas representadas en la simulación lleguen al borde de la pantalla, deben rebotar y continuar su movimiento.

Para lograrlo deberás utilizar funciones, condicionales, cambios de disfraces, eventos y valores aleatorios para simular el comportamiento del virus, empezando con una sola persona contagiada y observando cómo se propagan los contagios a medida que las personas interactúan.



Anexo 4.1. Inventario de bloques usados

Categoría	Bloque utilizado	Sesión	¿Para qué se usó?
Movimiento		4	Para que el objeto se ubicara en una posición al azar dentro del escenario.
Apariencia			

Categoría	Bloque utilizado	Sesión	¿Para qué se usó?
Eventos			
Control			
Sensores			

Categoría	Bloque utilizado	Sesión	¿Para qué se usó?
Operadores			
Variables			
Mis bloques			

TIC

Apoya:

Educación

BRITISH COUNCIL

Colombia Programa
{EL CÓDIGO A TU FUTURO}