

# Lección 1: Contraseña de Encendido

45 minutos

## Resumen



Esta lección fue creada originalmente por **Common Sense Education**.

Tener una contraseña segura y fuerte siempre es una muy buena idea. Sin embargo, ¿cómo podemos ayudar a los niños a crear mejores contraseñas y ser capaces de recordarlas? Usa los consejos en esta lección para ayudar a los niños a crear contraseñas seguras y fáciles de recordar.

## Propósito

Common Sense Education creó esta lección para enseñar a los estudiantes de qué manera las contraseñas fuertes pueden ayudar a proteger su privacidad.

## Estándares

Curso Completo Alineamiento

### Estándares de Ciencias de la Computación CSTA K-12 (2017)

- ▶ **NI** - Networks & the Internet

## Agenda

**Actividad inicial: pssst... ¿qué es una contraseña? (10 minutos)**

**Vocabulario Clave**

**Evaluar: ¡Oh, oh! si... entonces... (20 minutos)**

**Crear: refuerza tu contraseña (10 minutos)**

**Actividad de cierre: consejos para crear contraseñas (5 minutos)**

**Ampliación del aprendizaje**

**Niveles**

## Objetivos

Los estudiantes podrán:

- Definir la palabra “contraseña” y explicar su propósito.
- Practicar cómo crear una contraseña fuerte y fácil de recordar.

## Preparación

- Revisa el material instructivo.
- Imprime guías para cada estudiante.
- Prepara hojas o cuadernos para los estudiantes.

## Enlaces

**¡Aviso!** Por favor, haga una copia de cualquier documento que planea compartir con los estudiantes.

Para los profesores

- **Contraseña reforzada: Escenarios si... entonces...** - Guía de respuestas
- **Contraseña reforzada: cuestionario de la lección** - Guía de respuestas
- **¡Contraseña reforzada! diapositivas** - Diapositivas
- **Contraseña reforzada: refuerza tu contraseña** - Guía de respuestas

Para los estudiantes

- **Contraseña reforzada:** [Escenarios si... entonces...](#) - Guía
- **Contraseña reforzada:** [cuestionario de la lección](#) - Recurso
- **Contraseña reforzada:** [refuerza tu contraseña](#) - Guía

## Guía Didáctica

### Actividad inicial: pssst... ¿qué es una contraseña? (10 minutos)

#### Vocabulario Clave

- **Contraseña:** una secuencia secreta de letras, símbolos y números, que puedes usar para restringir el acceso a algo digital.
- **Frase:** un grupo de palabras que están juntas y son fáciles de recordar.
- **Símbolo:** un carácter que no es ni un número ni una letra, como #, ! o @.
- **Usuario:** un nombre que creas para registrarte en un sitio web, app o juego.

**Previo a la lección:** como actividad opcional previa a la lección, los estudiantes pueden jugar el juego [Protege la contraseña](#), en Digital Passport™ de Common Sense Education. Esta actividad ayudará a introducir conceptos clave de esta lección. Para más información, revisa la guía [Digital Passport Educator Guide](#).

**Pregunta:** ¿Hay algo en tu vida que proteges tomando resguardos?, ¿qué es?, ¿por qué y cómo lo proteges? ([diapositiva 4](#)).

Pide a los estudiantes que reflexionen-interactúen-expongan, y luego pide a algunos estudiantes que compartan sus respuestas. Las respuestas podrían ser: *esconder un diario, usar un candado para una bicicleta, mantener un juguete en un lugar seguro*, etc. Aclara que una de las maneras en que las personas protegen algo, es restringiendo quién puede acceder a ello.

**Pregunta:** *Pensemos en el uso de dispositivos, como un celular o un computador. ¿Cómo puedes limitar el acceso a algo que quieres proteger y por qué limitarías el acceso?*

Los ejemplos podrían ser: protección con contraseña, bloqueo del dispositivo, lector de huella digital, no permitir a otros utilizar el dispositivo.

**Comenta:** *Una de las formas más comunes para proteger los dispositivos y la información en línea es usar una contraseña. Una **contraseña** es una secuencia secreta de letras, símbolos y números, que puedes usar para restringir el acceso a algo digital. Sin embargo, algunas contraseñas son más fuertes que otras, ya que es más difícil que alguien pueda descifrarlas. Vamos a estudiar por qué las contraseñas fuertes son importantes y cómo puedes asegurarte de que la tuya sea fuerte ([diapositiva 5](#)).*

## Evaluar: ¡Oh, oh! si... entonces... (20 minutos)

**Distribuye** las **Guías de trabajo - Escenarios si... entonces....**. Comenta que harán una actividad “rompecabezas” para entender POR QUÉ son importantes las contraseñas.

**Divide** la clase en cinco grupos y coméntales que estos grupos se llamarán grupos “casa”. Pide a un estudiante que lea las instrucciones en voz alta (**diapositiva 6**).

**Asigna** a cada grupo uno de los escenarios de la guía de trabajo. Permite que los grupos trabajen de cinco a siete minutos.

**Vuelve a dividir** a los estudiantes en nuevos grupos de manera que cada grupo incluya al menos un estudiante por cada escenario (1, 2, 3, 4 y 5). Si es necesario, en un grupo de “expertos” puede haber más de un estudiante proveniente del mismo grupo “casa”. A cada “experto”, dale de cinco a siete minutos para que exponga su escenario y su respuesta, mientras el resto del grupo toma apuntes al respecto (**diapositiva 6**).

**Reúne** al curso y pregunta: *Basados en estos escenarios, ¿por qué creen que es importante tener una contraseña fuerte? Túrnense con un compañero para compartir sus ideas.* Permite que reflexionen-interactúen por un minuto.

Invita a los estudiantes a compartir sus respuestas. De ser necesario, sugiéreles referirse directamente a los escenarios y a las consecuencias de que la contraseña de alguien se vea comprometida. Los ejemplos podrían ser: perder dinero, información personal expuesta, robo de identidad, entre otras consecuencias futuras.

## Crear: refuerza tu contraseña (10 minutos)

**Comenta:** *Las consecuencias que acabamos de ver podrían parecer aterradoras. Pero hay algo que pueden hacer para asegurarse de que nadie pueda adivinar sus contraseñas, jamás. Estos son algunos pasos importantes para reforzar sus contraseñas.*

**Distribuye** y proyecta la **Guía de trabajo - Contraseña reforzada**. Di: *Una forma de crear una contraseña segura es comenzar con una frase fácil de recordar. Una **frase** es un grupo de palabras que están juntas y son fáciles de recordar. Vamos a hacer una ronda de práctica todos juntos, usando la frase “no pienso besar a un sapo”. Por lo tanto, ya completamos el paso uno, conseguir una frase (**diapositiva 7**).*

**Invita** a los estudiantes a leer el paso dos. Pregunta: *¿Qué palabra formaría nuestra frase de ejemplo?* Invítalos a responder y a añadir “npbaus” en la guía proyectada.

**Repite** los pasos tres, cuatro y cinco: lee cada paso en voz alta y pide a los estudiantes que respondan y completen la sección “Ronda de práctica” de la guía. Las respuestas de los estudiantes pueden variar, ya que pueden usar mayúsculas en distintas letras y agregar números diferentes. De ser necesario, aclara que pueden añadir los números en cualquier parte de la contraseña.

**Comenta:** *Ahora, seguirán estos pasos para crear sus propias contraseñas. Trabajen de forma individual para completar la tabla “Tu turno” en la guía.* Dale a los estudiantes cinco minutos para realizar la actividad.

## Actividad de cierre: consejos para crear contraseñas (5 minutos)

**Proyecta** los **consejos para crear contraseñas** y lee cada uno en voz alta. Indícales a los estudiantes que completen los espacios en blanco de sus guías mientras lees los consejos. Para el último consejo, aclara que un símbolo es *un carácter que no es ni un número ni una letra, cómo por ejemplo #, ! o @* (**diapositiva 8**).

**Comenta:** *A medida que crezcan, tener una contraseña segura será cada vez más importante. Las contraseñas los ayudarán a proteger sus perfiles en redes sociales cuando estén en el liceo, mantener sus notas en privado cuando estén en la universidad y proteger sus cuentas bancarias y sus cuentas comerciales cuando sean adultos.*

Haz que los estudiantes completen el **Cuestionario de la lección**.

## Ampliación del aprendizaje

Usa estas actividades para ampliar el aprendizaje de los estudiantes:

- **Actividad familiar**
- **Consejos para la familia**
- ¡Promueve un juego relacionado de BeakoutEDU que fue creado conjuntamente con Common Sense Education, llamado **Lip Sync Revenge!** Tendrás que crear una cuenta gratuita para poder acceder al juego y a los recursos relacionados.

## Niveles

Actividad sin cone...



Esta obra está disponible bajo una **Licencia Creative Commons (CC BY-NC-SA 4.0)**.

**Contáctanos** si desea contar con la licencia de los materiales de Code.org para uso comercial.

# Lección 2: Programación con Papel Cuadriculado

65 minutos

## Resumen

En esta lección de **context-setting**, los estudiantes usan símbolos para instruirse unos a otros en colorear cuadrados en papel cuadriculado. Al "programarse" unos a otros para hacer dibujos, los estudiantes tienen la oportunidad de experimentar algunos de los conceptos básicos de la programación de una manera divertida y accesible.

## Propósito

Los objetivos de esta actividad son desarrollar habilidades de pensamiento crítico, sembrar interés por este curso e introducir algunos conceptos fundamentales de la programación que serán usados a lo largo del curso. Al introducir conceptos básicos, como secuenciación o algoritmos, a través de una actividad sin conexión, incluso los estudiantes que no se sienten familiarizados con un computador podrán sentar las bases para comprender estos temas. En esta lección, los estudiantes aprenderán cómo desarrollar un algoritmo y codificarlo en un programa.

## Estándares

Curso Completo Alineamiento

### Estándares de Ciencias de la Computación CSTA K-12 (2017)

- ▶ **AP** - Algorithms & Programming

## Agenda

### Actividad previa (10 minutos)

#### Introducción

### Actividad Principal (30 minutos)

#### Programación con Papel Cuadriculado

### Actividad de cierre (15 minutos)

#### Reflexión

### Evaluación opcional (10 minutos)

#### Ampliación del aprendizaje

## Objetivos

Los estudiantes podrán:

- Explicar las limitaciones de traducir problemas desde el lenguaje humano al lenguaje de las máquinas
- Restructurar una secuencia de pasos en un programa codificado

## Preparación

- (Opcional) Mira el video **\*Lección en acción**.
- Imprime una **\*guía de trabajo** y una **\*evaluación** para cada estudiante.
- Asegúrate de que cada estudiante tenga su Diario de apuntes.

## Enlaces

**¡Aviso!** Por favor, haga una copia de cualquier documento que planea compartir con los estudiantes.

Para los profesores

- **Programación con Papel Cuadriculado** - Guía de respuestas de la guía de trabajo
- **Programación con Papel Cuadriculado** - Guía de respuestas de la evaluación

- **Programación con Papel Cuadrulado** - Video - Lección en acción

Para los estudiantes

- **Programación con Papel Cuadrulado** - Evaluación
- **Programación con Papel Cuadrulado** - Ficha de actividades
- **Programación con Papel Cuadrulado** - Video sin conexión (**Descarga**)

## Vocabulario

- **Algoritmo** - una lista de pasos para terminar una tarea.
- **Programa** - un algoritmo que se ha programado como algo que puede ser ejecutado por una máquina.

## Guía Didáctica

### Actividad previa (10 minutos)

#### Introducción

En esta actividad, los estudiantes programarán instrucciones para guiarse unos a otros a hacer dibujos, sin que el resto del grupo vea la imagen original. Esta actividad inicial contextualizará el ejercicio para la clase.

**Mostrar:** mira uno de los videos a continuación para contextualizar a los estudiantes sobre la clase de cosas que puede hacer un robot:

- **Robot Honda Asimo** (3:58)
- **Robot diseñador de huevos** (3:15)
- **Robot bailarín de Legos** (1:35)

**Análisis:** *¿Cómo creen que el robot sabe cómo hacer las cosas que hace? ¿Los robots tienen un cerebro similar al nuestro?*

Este breve análisis tiene como objetivo hacer énfasis en que los robots, a pesar de que pareciesen comportarse como humanos, realmente sólo responden a su programa. Es probable que los estudiantes hagan referencia a algunos robots de las películas y de la televisión con comportamientos más humanos. Guíalos a considerar robots que hayan visto u oído en la vida real, como el *Roombas*, o incluso asistentes digitales como *Amazon Alexa*.

Guía el análisis hacia una conversación sobre cómo las personas programan a los robots para hacer cosas específicas, a través de comandos específicos.

## Actividad Principal (30 minutos)

### Programación con Papel Cuadrulado

Actividad sin cone...

Programación con Papel Cuadrulado - Actividad Desconectada

En esta actividad, los estudiantes tomarán los roles de programador y robot. En una hoja de papel cuadrulado, pintarán cuadrados de acuerdo con los programas que se hayan escrito los unos a los otros.

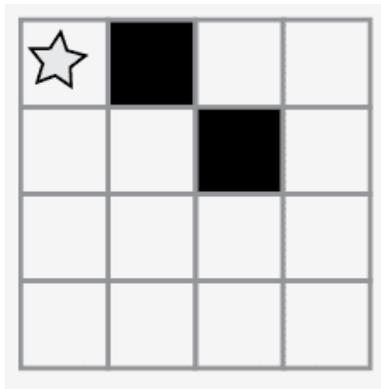
**Distribuir:** los estudiantes usarán una cuadrícula de 4x4 (u hojas de papel cuadrulado seccionado en cuadrículas de 4x4). También necesitarán una imagen de modelo.

**Mostrar:** proyecta estos comandos o escríbelos en la pizarra. No estarán mucho tiempo ahí, sólo lo suficiente para ayudar a los estudiantes a hacer la transición de algoritmos a programas.

Moverse un cuadrado a la derecha  
Moverse un cuadrado a la izquierda  
Moverse un cuadrado arriba  
Moverse un cuadrado abajo  
Rellenar el cuadrado

**Comenta:** *Hoy, vamos a programar robots... robots que ya están aquí en la sala! ¡Ustedes! Vamos a escribir programas usando símbolos con significados especiales, los que nos ayudarán a recrear imágenes. Primero, vamos a practicar todos juntos, como si yo fuera el robot y ustedes los programadores. Luego, vamos a dividirnos en grupos para que todos tengan su turno.*

**Mostrar:** muestra la imagen que usarás de ejemplo y la cuadrícula en blanco que rellenarás con tu Sistema de Ejecución Automática (SEA). Asegúrate de que las instrucciones, la cuadrícula y la imagen permanezcan visibles al mismo tiempo.



🎤 Observaciones

Aquí hay una imagen. Imaginen que soy un robot que funciona con un Sistema de Ejecución Automática (SEA). Esto significa que reaccionaré de forma automática a sus instrucciones, pero sólo a las que pueda entender.

Empiecen en la esquina superior izquierda. Guíen mi SEA diciéndome las instrucciones en voz alta.

**Modelar:** a continuación, la clase podría darte instrucciones como éstas. Cuando escuches una instrucción que pretendas seguir, asegúrate de repetir dicha instrucción en voz alta, de manera que los estudiantes puedan llevar un registro de tus movimientos.

Moverse un cuadrado a la derecha  
Rellenar el cuadrado  
Moverse un cuadrado a la derecha  
Moverse un cuadrado abajo  
Rellenar el cuadrado

Continúa con la actividad hasta completar la cuadrícula de ejemplo.

**Captar:** escribe cada uno de los comandos, de manera que los estudiantes puedan ver todos los pasos realizados para dibujar la imagen.

Moverse un cuadrado a la derecha  
Rellenar el cuadrado  
Moverse un cuadrado a la derecha  
Moverse un cuadrado abajo  
Rellenar el cuadrado

**Comenta:** Acaban de darme una lista de pasos para realizar una tarea. En programación, esto se llama algoritmo. Los algoritmos son geniales, ya que ustedes como programadores pueden entenderlos fácilmente. PERO, ¿qué pasa si queremos escribir el algoritmo para un dibujo como este?

**Mostrar:** muestra una imagen más complicada, como ésta.



A continuación, comienza a escribir algunas de las instrucciones para replicar la imagen. Con suerte, los estudiantes verán que escribir todo a mano podría fácilmente volverse una pesadilla.

Moverse un cuadrado a la derecha  
Rellenar el cuadrado  
Moverse un cuadrado a la derecha  
Moverse un cuadrado a la derecha  
Rellenar el cuadrado  
Moverse un cuadrado abajo  
Moverse un cuadrado a la izquierda  
Rellenar el cuadrado  
Moverse un cuadrado a la izquierda  
Moverse un cuadrado a la izquierda  
Rellenar el cuadrado  
¡Y 12 instrucciones más!

\*\*Pantalla: \*\* Muestra a los alumnos esta lista de símbolos.

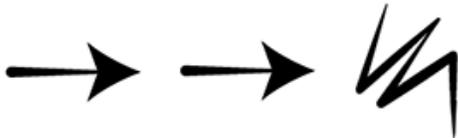


**Análisis:** ¿cómo podemos usar estos símbolos para facilitar nuestras instrucciones?

Dibuja ideas que apoyen la transición de instrucciones verbales a símbolos. Una vez que los estudiantes comprendan la idea, indica que el texto:

- “Moverse un cuadrado a la derecha, moverse un cuadrado a la derecha, rellenar el cuadrado”

Ahora corresponde al programa:



El objetivo de este análisis es llegar a la idea de que los estudiantes pueden usar símbolos para representar frases completas. Una vez que lo comprendan, coméntales que pasar de enlistar pasos detallados a codificarlos, se llama "**programación**".

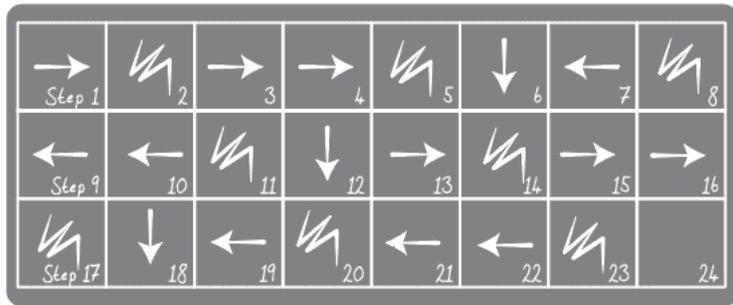
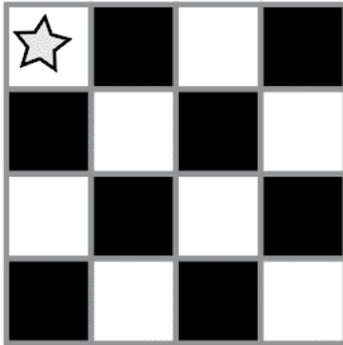
**Modelar:** ahora, que la clase te ayude a dibujar la imagen grande usando sólo símbolos. Por ahora, no te preocupes si se incluyen pasos innecesarios; si el programa funciona para recrear la imagen, considéralo correcto.

💡 Consejo didáctico ▲

Nota que hemos escrito el programa de izquierda a derecha, como se leería un libro en español. Algunos estudiantes prefieren este método, mientras que otros prefieren empezar cada línea de la cuadrícula en una nueva línea de la hija. La forma en que escriban el programa no importa mucho, mientras los demás estudiantes puedan leer y seguir el programa.

En este punto, los estudiantes podrían estar emocionados y animados dando sugerencias. Si entienden lo esencial del ejercicio, este es un buen momento para analizar otras alternativas para rellenar la misma cuadrícula. Si aún no están listos, guarda esta idea para otro día y realiza otro ejemplo.

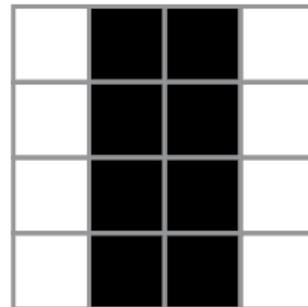
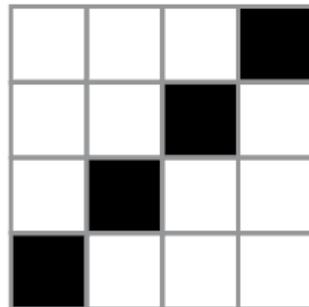
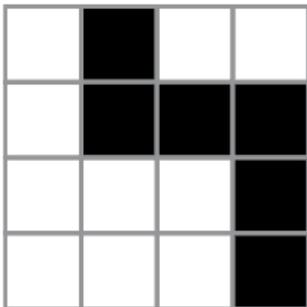
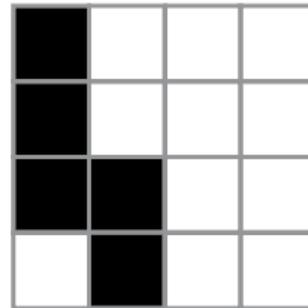
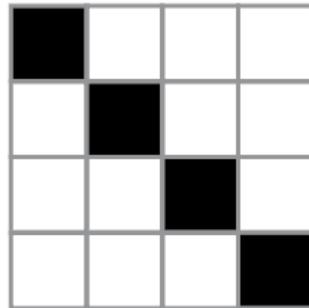
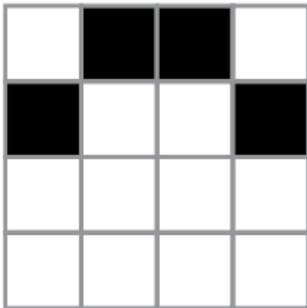
Consulta una solución de ejemplo a continuación:



## Turno del estudiante

**Grupo:** divide a los estudiantes en parejas o pequeños grupos.

- Que cada grupo/pareja elija una imagen de la guía de trabajo.
- Discutan con sus compañeros el algoritmo necesario para dibujar la imagen elegida.
- Conviertan el algoritmo en un programa, usando los símbolos.
- Intercambien los programas con otros grupos/parejas, para que dibujen las imágenes de otros.
- Elijan otra imagen y involvamos a empezar!



## Actividad de cierre (15 minutos)

### Reflexión

#### Sugerencias:

- ¿Sobre qué se trataba la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- Dibuja otra imagen que podrías codificar. ¿Puedes escribir el programa que corresponde con la imagen?
- ¿Qué otro tipo de robots podríamos programar si cambiáramos lo que significan las flechas?

💡 Consejo didáctico ▲

Las actividades de reflexión como estas se pueden completar de diversas formas. ¡Escribir un diario o debatir (con grupos o compañeros) son excelentes opciones!

## Evaluación opcional (10 minutos)

- Entrega la **\*Evaluación - Programación con papel cuadriculado**. Luego de haber explicado claramente las instrucciones, permite que los estudiantes realicen la evaluación de forma individual.
- Gracias a la actividad previa, esto no debería significar ningún problema para ellos.

## Ampliación del aprendizaje

Usa estas actividades para ampliar el aprendizaje de los estudiantes. Se pueden usar como actividades extras fuera del aula.

#### Cada vez mejor

- Haz que tu clase intente hacer sus propias imágenes.
- ¿Pueden descifrar cómo codificar las imágenes que acaban de crear?

#### Desafío de curso

- Dibuja una imagen en una cuadrícula de 5x5.
- ¿La clase puede codificarla contigo?



Esta obra está disponible bajo una **Licencia Creative Commons (CC BY-NC-SA 4.0)**.

**Contáctanos** si desea contar con la licencia de los materiales de Code.org para uso comercial.

# Lección 3: Introducción a los desafíos en línea

60 minutos

## Resumen

En esta lección de **desarrollo de habilidades**, los estudiantes practicarán sus habilidades *secuenciación* y depuración en rompecabezas de laberinto.

## Propósito

Comprendemos que en cada sala de clases hay un espectro de comprensión diferente para cada tema. Algunos estudiantes pueden ser muy hábiles con los computadores mientras que otros pueden tener muy poca experiencia con ellos. Con el objetivo de nivelar el área de juego (y de aprendizaje), hemos desarrollado esta “etapa de reforzamiento” para el curso D. Esto puede ser tanto una introducción como un repaso de cómo usar Code.org y conceptos básicos de las ciencias de la computación. Esta lección cubre todos los prerrequisitos requeridos para comenzar el curso D.

## Estándares

Curso Completo Alineamiento

### Estándares de Ciencias de la Computación CSTA K-12 (2017)

- ▶ **AP** - Algorithms & Programming

## Agenda

### Actividad previa (10 minutos)

#### Introducción

### Actividad puente: programación (10 minutos)

#### Revisión de los desafíos en línea

### Actividad Principal (30 minutos)

#### Desafíos en línea

### Actividad de cierre (10 minutos)

#### Reflexión

## Objetivos

Los estudiantes podrán:

- Descomponer una secuencia larga de instrucciones en una secuencia de repetición más corta.
- Modificar un programa existente para resolver errores.
- Ordenar comandos de movimiento en pasos secuenciales dentro de un programa.

## Preparación

- Juega con los rompecabezas para encontrar cualquier área potencialmente problemática para tu clase.
- Asegúrate de que cada estudiante tenga un diario de reflexión.

## Enlaces

**¡Aviso!** Por favor, haga una copia de cualquier documento que planea compartir con los estudiantes.

Para los estudiantes

- **Programación en parejas** - Video del estudiante

## Vocabulario

- **Bug (error)** - parte de un programa que no funciona correctamente.
- **Depuración** - (v) encontrar y solucionar problemas en un algoritmo o programa.
- **Bucle** - la acción de hacer algo una y otra vez.
- **Programa** - un algoritmo que se ha programado como algo que puede ser ejecutado por una máquina.
- **Programación** - El arte de crear un programa

## Guía Didáctica

### Actividad previa (10 minutos)

#### Introducción

Los estudiantes aprenderán muchos conceptos nuevos o repasarán muchos conceptos básicos. En base a la experiencia de tu clase, puedes revisar el vocabulario a continuación o pasar directamente a la actividad puente. Si las definiciones no están cubiertas explícitamente, recomendamos usar las palabras en oraciones.

#### Vocabulario

Esta lección tiene cuatro nuevas e importantes palabras:

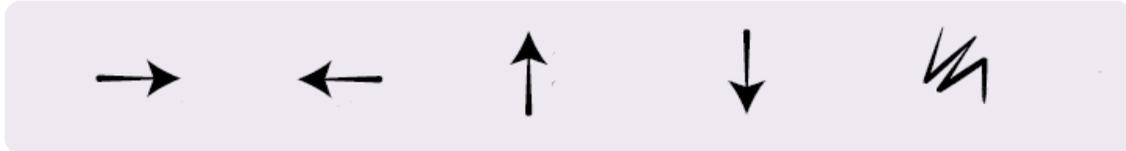
- **Programa:** *díganlo conmigo, PRO - GRA - MA.*  
Un algoritmo que ha sido codificado de forma que pueda ser ejecutado por una máquina.
- **Programación:** *díganlo conmigo, PRO - GRA - MA - CIÓN.*  
El arte de crear un programa.
- **Bug** - *\_díganlo conmigo BAA - G.*  
Parte de un programa que no funciona correctamente.
- **Depurar:** *díganlo conmigo DE - PU - RAR.*  
Encontrar y solucionar los problemas en un algoritmo o programa.
- **Bucle:** *díganlo conmigo BU - CLE.*  
La acción de hacer algo una y otra vez.

## Actividad puente: programación (10 minutos)

Esta actividad ayudará a tus estudiantes a traer los conceptos sin conexión de “Programación con papel cuadriculado” al mundo digital al cual se están sumergiendo.

### Revisión de los desafíos en línea

Elige un desafío de la lección. Para esta actividad recomendamos el desafío 6. Divide a los estudiantes en grupos de tres o cuatro. Pídeles que “programen” a Red, de Angry Birds, para que llegue al cerdo, usando las flechas de “Programación con papel cuadriculado”.



La clase no necesitará usar el último símbolo.

Una vez que toda la clase tenga una respuesta, discutan el camino a seguir en conjunto.

## Actividad Principal (30 minutos)

### Desafíos en línea

💡 Consejo didáctico ▲

Enseña a los estudiantes la manera correcta de ayudar a sus compañeros:

- No sentarse en la silla del compañero
- No usar el teclado del compañero
- No tocar el mouse del compañero
- Asegurarte de que tu compañero pueda describirte la solución en voz alta antes de que te vayas

Los docentes son de vital importancia en la educación de las ciencias de la computación y juegan un rol fundamental para producir un ambiente vibrante y colaborativo en la sala de clases. Durante las actividades en línea, el rol del docente es alentar y apoyar. Los desafíos en línea están estructurados para estar centrados en el estudiante, así que los docentes deben evitar involucrarse cuando los estudiantes tengan problemas para resolverlos. Algunas ideas de cómo hacerlo son:

- Usa la programación en parejas cada vez que sea posible durante la actividad.
- A través de desafíos o preguntas, anima a los estudiantes para que busquen respuestas con sus respectivas parejas.
- Las preguntas sin responder pueden ser delegadas a un grupo cercano, que podría ya tener la respuesta.
- Pide a los estudiantes que describan el problema que estén viendo. ¿Qué se supone que debe hacer?, ¿qué hace?, ¿qué te dice eso?
- Recuérdales que la frustración es un paso en el camino del aprendizaje y que la perseverancia dará sus frutos.
- Si un estudiante sigue con problemas para avanzar después de las acciones anteriores, haz preguntas clave para que los estudiantes identifiquen el error por ellos mismos.

1

Vídeo: Introducción al Laberinto: Programación con Bloques

2-7

Desarrollo de Habilidades

2

3

4

5

6

7

8

Desafío

9-10

Práctica

9

10

## Actividad de cierre (10 minutos)

### Reflexión

#### Sugerencias:

- ¿Sobre qué se trataba la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- Haz una lista de los errores que encontraste en tus programas hoy.
- ¿Cuál fue tu desafío favorito? Dibuja a tu personaje favorito completando desafíos.



Esta obra está disponible bajo una **Licencia Creative Commons (CC BY-NC-SA 4.0)**.

**Contáctanos** si desea contar con la licencia de los materiales de Code.org para uso comercial.

# Lección 4: Programación de relevos

50 minutos

## Resumen

Esta actividad de **contexto** comenzará con una breve lección sobre la depuración y la constancia, y luego pasará rápidamente a una carrera contrarreloj en la que los alumnos se dividirán en equipos y trabajarán juntos para escribir la instrucción de un programa a la vez.

## Propósito

El trabajo en equipo es muy importante en las ciencias de la computación. Los equipos escriben y depuran códigos en conjunto, en lugar de trabajar de forma individual. En esta lección, los estudiantes aprenderán a trabajar en equipo intentando ser lo más eficientes posible.

Esta actividad también involucra el concepto de urgencia, lo cual enseñará a los estudiantes que deben administrar su tiempo cuidadosamente, evitando cometer errores sin comprometer mucho tiempo. Esta experiencia puede ser estresante (¡y lo será!). Asegúrate de compartir con los estudiantes las herramientas para lidiar con la frustración.

## Estándares

Curso Completo Alineamiento

### Estándares de Ciencias de la Computación CSTA K-12 (2017)

- ▶ **AP** - Algorithms & Programming

## Agenda

### Actividad previa (15 minutos)

#### ¿Dónde me equivoqué?

### Actividad Principal (20 minutos)

#### Programación de relevos

### Actividad de cierre (15 minutos)

#### Reflexión

### Ampliación del aprendizaje

## Objetivos

Los estudiantes podrán:

- Definir ideas usando código y símbolos.
- Identificar las señales de frustración.
- Verificar el trabajo hecho por compañeros.

## Preparación

- Mira el **\*Video del docente**.
- Encuentra un espacio abierto para esta actividad, como el gimnasio o un área verde.
- Imprime un **\*Paquete de actividades** para cada grupo.
- Provee a cada grupo hojas y lápices.
- Imprimir una **\*Guía de trabajo** para cada estudiante.
- Asegúrate de que cada estudiante tenga su Diario de apuntes.

## Enlaces

**¡Aviso!** Por favor, haga una copia de cualquier documento que planee compartir con los estudiantes.

Para los profesores

- **Programación de relevos** - Guía de respuestas de la guía de trabajo

▼ Hacer una copia

- **Programación de relevos** - Imagen de depuración para el docente
- **Programación de relevos** - Video para el docente

Para los estudiantes

- **Programación de relevos** - Paquete de actividades
- **Programación de relevos** - Guía de trabajo
- **Programación de relevos** - Video sin conexión  
([Descarga](#))

## Vocabulario

- **Algoritmo** - una lista de pasos para terminar una tarea.
- **Bug (error)** - parte de un programa que no funciona correctamente.
- **Depuración** - (v) encontrar y solucionar problemas en un algoritmo o programa.
- **Frustración** - Sentirse molesto o enojado porque algo no funciona como quieres.
- **Perseverancia** - intentar una y otra vez, incluso cuando algo es muy difícil.
- **Programa** - un algoritmo que se ha programado como algo que puede ser ejecutado por una máquina.

## Guía Didáctica

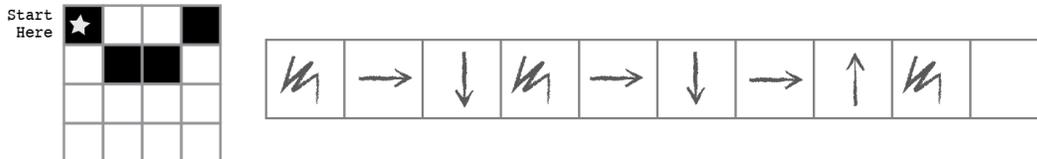
### Actividad previa (15 minutos)

Recordemos que en “Programación con papel cuadriculado” guiamos el Sistema de Ejecución Automática (SEA) de nuestros compañeros usando flechas. Esta actividad inicial trae de vuelta esas ideas, las cuales serán necesarias en la actividad principal.

## ¿Dónde me equivoqué?

**Objetivo:** en esta lección, queremos ayudar a los estudiantes a identificar y resolver bugs en sus programas. La forma más sencilla de hacerlo es trabajar en un programa que ya contenga bugs, pero que no sean culpa de los estudiantes. Una vez que te hayan ayudado a reparar “tu” programa, comparte con ellos lo frustrante que puede llegar a ser cometer errores, y ayúdales a ver que esos sentimientos son absolutamente normales, no deben sentirse avergonzados por ellos.

**Mostrar:** muestra la imagen de la **\*Guía de depuración para el docente.**



**Análisis:** llama la atención de la clase y diles que estás estancado! Tienes este desafío y crees haberlo resuelto, pero parece que no funciona bien. Tu programa tiene un bug, ¿pueden ayudarlo a repararlo?

Tómate un momento para repasar las reglas:

- Comenzar en la estrella.
- Seguir las instrucciones paso a paso.
- Terminar cuando se hayan rellenado todos los cuadrados de la derecha.

*Opcional:* continúa rellenando los cuadrados en una cuadrícula en blanco. Expresa tu frustración cuando la imagen no resulte de la forma en que querías.

**Reflexión:** ¿pueden descifrar por qué mi programa no funciona?

**Interacción:** pide a los estudiantes que trabajen en equipo para ver si pueden descifrar cómo debería estar escrito el programa.

**Exposición:** pregunta si alguien encontró una manera de resolver el problema. Cuando te den una respuesta correcta, idiles a los estudiantes que son muy buenos “depurando”!

**Análisis:** pregunta si lograron notar cómo te sentías cuando no podías encontrar la respuesta. Podrían sugerir que te sentías “triste” o “molesto”. En lugar de decirles que “no”, explica que te sentías un poco molesto, un poco triste, un poco confundido. Cuando se juntan todas esas sensaciones, nace el sentimiento de “frustración”. Cuando estás “frustrado”, quizás puedas pensar que estás molesto, triste o confundido. Quizás te veas tentado a rendirte; sin embargo, la frustración es un sentimiento normal y es un gran indicio de que estás a punto de aprender algo. En lugar de rendirte, intenta ser perseverante. Sigue intentando una y otra vez. Después de algunos intentos, ¡empezarás a entender cómo depurar tus problemas!

### Rincón de contenido

Sugerencias para lidiar con la frustración:

- Contar hasta 10
- Respirar hondo
- Escribir acerca del programa
- Hablar con algún amigo sobre el problema
- Pedir ayuda

### Sugerencias para ser perseverante

- Llevar registro de lo que ya has intentado
- Describir qué es lo que pasa
- Describir qué debería hacer
- ¿Qué te dice eso?
- Haz un cambio y vuelve a intentar

**Distribuir:** para asegurarte de que los estudiantes entienden la idea de encontrar y solucionar errores (depurar), distribuye la **\*Guía de trabajo** y que trabajen en parejas.

*Opcional:* si no quieres invertir mucho tiempo en esta actividad, pueden realizarla en conjunto como clase.

**Transición:** ¡llegó la hora de la acción!

## Actividad Principal (20 minutos)

### Programación de relevos

Actividad sin cone...

Programación de Relés - Actividad Desconectada

Con los conceptos de “Programación con papel cuadrulado” en mente, ¡es momento de dividirse en equipos y prepararse para la actividad en relevos!

**Preparación:** prepárate imprimiendo el **\*Paquete de actividades** para cada equipo, de 4 a 5 estudiantes. Corta o dobla cada página por la línea punteada.

#### Consejo didáctico

Algunas cosas deben ser aclaradas y recordadas de vez en cuando:

- Solo una persona de cada grupo puede estar frente a la imagen a la vez.
- Está permitido discutir el algoritmo con el resto del grupo mientras están en la fila, incluso pueden planificar quién y qué van a escribir en el programa.
- Cuando un estudiante depura el programa tachando una instrucción incorrecta (o un conjunto de instrucciones incorrectas), cuenta como su turno. El siguiente estudiante debe descifrar cómo corregir las instrucciones eliminadas.

Revisa las reglas del juego con tu clase:

- Dividirse en equipos de 3-5 estudiantes.
- Que cada grupo haga una fila, uno detrás de otro.
- Colocar una de las imágenes guía al otro lado de la sala/gimnasio/área para cada equipo (usa la misma imagen para todos los equipos).
- El primer estudiante de la fila debe correr hasta la imagen, revisarla y escribir el primer símbolo del programa para reproducir esa imagen.
- El primer estudiante debe volver y tocar al siguiente de la fila. Luego, debe ponerse al final de la fila.
- La siguiente persona en la fila debe correr hasta la imagen y revisarla. Luego, debe revisar el programa que se ha escrito y realizar sólo una de estas dos opciones: depurar el programa

tachando los símbolos incorrectos o agregar un símbolo nuevo. El estudiante luego corre de vuelta y toca al siguiente en la fila para que sea su turno. Repetir hasta que un equipo termine el programa.

¡El primer grupo que consiga un programa que coincida con la imagen es el ganador! Repite esta actividad varias veces, aumentando la dificultad.

Repite el juego tantas veces como puedas, hasta que se agote el tiempo o hasta que los estudiantes se sientan fatigados.

**Transición:** una vez que el juego acabe, reúne a todos los estudiantes en un círculo para compartir lo aprendido.

**Debate:** *¿qué aprendimos hoy?*

- *¿Qué tal si cada persona en la fila pudiese hacer 5 flechas a la vez?*
  - *¿Qué tan importante puede ser depurar nuestro propio trabajo y el trabajo del programador anterior a nosotros?*
  - *¿Y si pudiesen hacerse 10 flechas?*
  - *¿Y si fuesen 10.000? ¿Sería más o menos importante?*
- *¿Crees que un programa es mejor o peor cuando más de una persona trabaja en él?*
- *¿Crees que las personas comenten más o menos errores cuando están apurados?*
- *Si encuentras un error, ¿debes deshacer todo el programa y comenzar desde cero?*

## Actividad de cierre (15 minutos)

### Reflexión

**Sugerencias:**

- ¿Sobre qué se trataba la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- ¿Cómo es que el trabajo en equipo influyó en el éxito de escribir el programa de hoy?
- ¿Te sentiste frustrado en algún punto? ¿qué hiciste respecto a eso?

## Ampliación del aprendizaje

Usa estas actividades para ampliar el aprendizaje de los estudiantes. Se pueden usar como actividades extras fuera del aula.

**Pasa la hoja**

- Si no tienes tiempo o espacio para una actividad de relevos al aire libre, puedes reunir a los estudiantes en grupos y que pasen la hoja con el programa de un puesto a otro. Haz que cada estudiante haga una flecha antes de pasar la hoja a un compañero.

**Liénalo, muévelo**

- Como docente, dibuja una imagen con tantos cuadrados pintados como niños haya en cada grupo.
- Pide a los estudiantes que anoten tantas flechas en el programa como sea necesario para llegar a un cuadrado pintado, incluyendo la acción de pintar ese cuadrado, antes de pasar la hoja al siguiente estudiante.

**Depurando juntos**

Dibuja una imagen en la pizarra. Pide a cada estudiante crear un programa para esa imagen. Luego, haz que intercambien sus programas con sus compañeros de al lado y depuren sus códigos.

- Encierra en un círculo el primer paso incorrecto, luego devuelve el programa.
- Dales a los estudiantes otra oportunidad para revisar y depurar sus propios programas.
- Pide que un voluntario comparta su programa.

Pregunta:

- *¿Cuántos tienen el mismo programa?*
- *¿Alguien tiene algo diferente?*



Esta obra está disponible bajo una **Licencia Creative Commons (CC BY-NC-SA 4.0)**.

**Contáctanos** si desea contar con la licencia de los materiales de Code.org para uso comercial.

# Lección 5: Depuración con Laurel

75 minutos

## Resumen

En esta actividad online, los/as estudiantes practicarán debugging en el entorno "recolector". Los/as estudiantes podrán practicar la lectura y edición de código para arreglar desafíos con algoritmos simples, loops y loops anidados.

## Propósito

El propósito de esta lección es enseñar a los estudiantes que fallar es normal cuando estamos aprendiendo nuevas habilidades. A los estudiantes se les darán programas que NO funcionan correctamente y se les pedirá repararlos. Este proceso, llamado "depuración", los ayudará a desarrollar habilidades de pensamiento crítico y resolución de problemas, habilidades que los acompañarán mientras avanzan a proyectos de programación cada vez más complejos.

## Estándares

Curso Completo Alineamiento

### Estándares de Ciencias de la Computación CSTA K-12 (2017)

- ▶ **AP** - Algorithms & Programming

## Agenda

### Actividad previa (15 minutos)

#### Introducción

### Actividades puente: elige una (15 minutos)

#### Actividad sin conexión: bloques de papel

#### Revisión de los desafíos en línea

### Actividad Principal (30 minutos)

#### Depuración con Laurel

### Actividad de cierre (15 minutos)

#### Reflexión

## Objetivos

Los estudiantes podrán:

- Describir e implementar un plan para depurar un programa.
- Identificar un bug y los problemas que causa en un programa.
- Leer y comprender un código dado.

## Preparación

- Juega con los rompecabezas para encontrar cualquier área potencialmente problemática para tu clase.
- Asegúrate de que cada estudiante tenga un diario de reflexión.

## Enlaces

**¡Aviso!** Por favor, haga una copia de cualquier documento que planee compartir con los estudiantes.

Para los estudiantes

- **Bloques sin conexión (cursos C-F)** - Manipulativos
- **Programación de relevos** - Paquete de actividades

## Vocabulario

- **Bug (error)** - parte de un programa que no funciona correctamente.
- **Depuración** - (v) encontrar y solucionar problemas en un algoritmo o programa.

## Guía Didáctica

### Actividad previa (15 minutos)

#### Introducción

Una de las partes más importantes en el aprendizaje de la programación es aprender a depurar. Pregunta a la clase si alguna vez han aprendido una habilidad nueva y si alguna vez han fallado.

Por ejemplo:

- Aprender a andar en bicicleta y caerse.
- Aprender a hornear y quemar la comida.
- Aprender algún deporte y no ganar ningún juego.

Equivocarse es muy común cuando se aprenden cosas nuevas. Pide a los estudiantes que discutan sobre situaciones en que se hayan equivocado y cómo las superaron.

En la programación, los programadores normalmente encuentran “bugs” en sus códigos.

- **Bug (error)**: parte de un programa que no funciona correctamente.

Un bug puede realmente arruinar un programa, por lo que es importante aprender a “depurar” el código.

- **Depuración**: encontrar y solucionar los problemas en un algoritmo o programa.

Continúa la conversación si crees que tu clase necesita una mayor introducción, pero no olvides dejar tiempo para una de las actividades puente.

### Actividades puente: elige una (15 minutos)

Estas actividades ayudarán a tus estudiantes a traer los conceptos sin conexión de “Depuración sin conexión: programación de relevos” al mundo digital al cual se están sumergiendo. Elige una de las siguientes actividades:

#### Actividad sin conexión: bloques de papel

Divide a la clase en equipos de 3 a 5 estudiantes y encuentra un lugar amplio, puede ser un gimnasio o un área verde. Ordena los equipos en fila, similar a como se hizo en la actividad “Programación de relevos”. Toma un diseño mediadamente difícil del **\*Paquete de actividades**. A una distancia considerable, coloca una imagen por cada equipo. Además, entrégales suficientes bloques de papel de

los **\*Manipulativos**. Cada equipo necesitará muchos bloques `llenar 1` y `mover__`. Los bloques `mover__` pueden ser rellenados con anticipación o durante el mismo juego. De cualquier modo, asegúrate de que los bloques estén bien definidos mientras se desarrolla el juego.

Una vez que todos los equipos estén formados, muestra las reglas o léelas en voz alta:

- El primer estudiante en la fila debe correr hacia la imagen, revisarla, y colocar el primer bloque de código en el programa para reproducir esa imagen.
- El primer estudiante debe volver y tocar al siguiente de la fila. Luego, debe ponerse al final de la fila.
- La siguiente persona en la fila debe correr hasta la imagen y revisarla. Luego, debe revisar el programa que se ha escrito y hacer una de dos: depurar el programa quitando los bloques de código incorrectos o agregar un bloque nuevo. Luego, el estudiante debe correr de vuelta y toca al siguiente en la fila para que sea su turno. Repetir hasta que un equipo termine el programa.

Asegúrate de que los estudiantes usen sólo los bloques `llenar 1` y `mover__` y que sólo coloquen un bloque por turno. ¡El primer equipo que escriba *correctamente* el código de su imagen, gana!

-O-

## Revisión de los desafíos en línea

Agrupar a los estudiantes en grupos de 3. Elige un desafío de la lección, recomendamos el número 5. Pídeles que se sienten frente a un computador con el desafío en pantalla. Cada equipo puede tener sólo un computador y un estudiante mirando la pantalla. Muestra o lee las reglas:

- Sólo un estudiante por equipo puede estar mirando la pantalla.
- Esa persona sólo puede añadir o borrar un bloque a la vez. Cuando esa persona haya añadido o eliminado un bloque, puede tocar el hombro de un compañero para que sea su turno.
- El siguiente estudiante puede ponerse frente a la pantalla y jugar su turno.
- Los turnos no pueden repetirse ni saltarse, todos deben jugar la misma cantidad de veces.

¡El primer equipo en resolver el desafío *correctamente*, gana!

## Actividad Principal (30 minutos)

### Depuración con Laurel

Podría ser muy útil que los integrantes de los equipos de la actividad puente se sienten cerca unos de otros. Cada estudiante debe realizar los desafíos de forma individual o en parejas; sin embargo, contar con un grupo de trabajo conocido para hacer y responder preguntas puede ayudarlos a sentir más confianza y comprender mejor el tema.



Vídeo: Depuración con el Botón Step



Desarrollo de Habilidades

2

3

4

5

6

7

Desafío

8-9

Práctica

8

9

## Actividad de cierre (15 minutos)

### Reflexión

El acto de escribir en sus diarios sobre lo aprendido, respecto de si les pareció útil y de lo que sintieron, ayuda a tus estudiantes a fortalecer cualquier conocimiento que hayan obtenido hoy y sirve como un resumen al que puedan recurrir en el futuro.

#### Sugerencias:

- ¿Sobre qué se trataba la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- ¿Qué es un bug? ¿Cómo sabes que hay un bug en tu programa?
- ¿Qué significa depurar un código? ¿Cómo depuras un programa?



Esta obra está disponible bajo una [Licencia Creative Commons \(CC BY-NC-SA 4.0\)](https://creativecommons.org/licenses/by-nc-sa/4.0/).

[Contáctanos](#) si desea contar con la licencia de los materiales de Code.org para uso comercial.

# Lección 6: Eventos de Rebote

50 minutos

## Resumen

En esta lección de **context-setting/building** skill-building\*\*, los alumnos aprenderán qué son los eventos y cómo los programadores los usan en los videojuegos. Los alumnos construirán un juego que podrán personalizar con diferentes velocidades y sonidos.

## Propósito

Los eventos son muy comunes en los programas de computador, especialmente en los videojuegos.

En esta lección, los estudiantes desarrollarán la comprensión de los eventos al desarrollar un juego de deportes. Los estudiantes aprenderán a hacer que su raqueta se mueva al presionar las flechas y a producir sonidos cuando los objetos colisionen. Al final, podrán personalizar su juego para hacerlo aún más único!

## Estándares

Curso Completo Alineamiento

### Estándares de Ciencias de la Computación CSTA K-12 (2017)

- ▶ **AP** - Algorithms & Programming
- ▶ **CS** - Computing Systems

## Oportunidades multidisciplinarias

### Estándares Científicos de la Próxima Generación

- ▶ **ETS** - Engineering in the Sciences

## Agenda

### Actividad previa (10 minutos)

#### Introducción

### Actividad Principal (30 minutos)

#### Eventos de Rebote

### Actividad de cierre (10 minutos)

#### Reflexión

#### Cómo funcionan los eventos

### Ampliación del aprendizaje

## Objetivos

Los estudiantes podrán:

- Crear un juego interactivo mediante secuencia y controladores de eventos.
- Identificar acciones que se correlacionan con eventos de entrada.
- Compartir un artefacto creativo con otros estudiantes.

## Preparación

- Asegúrate de que cada estudiante tenga su Diario de apuntes.
- Realiza los desafíos para encontrar cualquier área potencialmente problemática para tu clase.
- Lee las **\*diapositivas** (para mostrárselas a los estudiantes al final de la clase).

## Enlaces

**¡Aviso!** Por favor, haga una copia de cualquier documento que planea compartir con los estudiantes.

Para los profesores

- **Eventos en el Rebote - ¿Cómo se producen?**
- **Niveles de programación abiertos** - Guía de respuestas

## Oportunidad multidisciplinaria

Para los estudiantes

- **Momento de reflexión en línea** - Vídeo

## Vocabulario

- **Evento** - una acción que hace que algo suceda.

## Guía Didáctica

### Actividad previa (10 minutos)

#### Introducción

**Comenta:** *Veamos si podemos encontrar un patrón aquí:*

- *Cuando pulsas un interruptor, las luces se encienden.*
- *Cuando tocas un dispositivo, se inicia una app.*
- *Cuando el balón toca la red, anotas un punto.*
- *Cuando (evento), (acción).*

*En ciencias de la computación, los eventos provocan otras acciones. Nuestra nueva palabra de vocabulario de hoy es evento. Normalmente, cuando oímos "evento", pensamos en:*

- Excursión
- Competición
- Fiesta de cumpleaños

*Hoy vamos a aprender un nuevo significado para la palabra "evento". Vamos a centrarnos en los "eventos" que provocan otras acciones, como cuando al meter el balón en la meta se anota un punto o al pulsar un botón se mueve un personaje en un juego.*

Esta lección tiene una nueva e importante palabra:

**Evento:** *díganlo conmigo, E - ven - to.*

Una acción que causa que algo pase.

### Actividad Principal (30 minutos)

#### Eventos de Rebote



1-7

Desarrollo de Habilidades

1

2

3

4

5

6

7

## Desafíos en línea

Al final de estos desafíos, los estudiantes tendrán la oportunidad de personalizar sus juegos. Pídeles que busquen formas de hacer sus juegos aún más desafiantes. Por ejemplo, intentar jugar con varias pelotas a la vez o que cada vez que una pelota rebote en una pared, aparezca una nueva pelota.

### Consejo didáctico

Recuerda a los estudiantes que sólo deben compartir su trabajo con sus amigos o familiares cercanos. Para más información, visualiza o muestra a la clase el video **\*Momento para reflexionar en línea.**

## Actividad de cierre (10 minutos)

### Reflexión

#### Sugerencias:

- ¿Sobre qué se trataba la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- ¿Qué hiciste para que tu juego fuese súper genial?
- ¿Qué clase de juego te gustaría programar en el futuro?

## Cómo funcionan los eventos

Muestra las diapositivas **\*Eventos de rebote ¿cómo funcionan?** a los estudiantes. Primero, permíteles anotar el orden correcto de los **eventos** de la segunda diapositiva en sus diarios. Llama a algunos estudiantes para que compartan sus respuestas antes de revelar la tercera diapositiva. Discutan la secuencia correcta con la clase.

### Consejo didáctico

A pesar de ser introducidas en esta lección, las diapositivas **\*Eventos de rebote ¿cómo funcionan?** pueden aplicarse de forma más general para expresar la relación entre hardware y software. En particular, la última diapositiva simplifica la secuencia de entrada y salida (input y output) y puede usarse como un póster para la sala de clases.

## Ampliación del aprendizaje

### Vamos a jugar con la pelota

Sal con los estudiantes al aire libre para jugar algún juego de pelota. Lleven registro de los eventos y las acciones. Por ejemplo, si en el baloncesto no se hace rebotar la pelota, se comete una falta de desplazamiento y se debe pasar la pelota al equipo contrario. En el fútbol, patear la pelota fuera del área resulta en que el otro equipo tiene un saque de costado. ¡Meter la pelota en el arco resulta en un gol! Si los estudiantes están interesados, ¡inventa más eventos! Que todos los estudiantes griten “¡YUPIII!” cuando el capitán de un equipo haga un gol, o que todos los estudiantes se tiren al piso y rueden si un estudiante hace dos goles seguidos.

# Oportunidad multidisciplinaria

Comentarios del día del juego (45-60 minutos)

 **Ciencias de la computación + Lengua y Literatura + Matemáticas + Ciencia**

**Comentarios del día del juego** es una actividad opcional alineada con las normas de Lengua y Literatura, Matemáticas y Ciencia de Common, escrita por nuestra comunidad docente. En esta actividad, los estudiantes leerán las instrucciones en voz alta a medida que aprenden a programar con eventos de rebote para crear un juego. Una vez finalizado el juego, los estudiantes harán comentarios jugada por jugada mientras sus compañeros de clase predicen en qué dirección rebotará la pelota para anotar. Los estudiantes también completarán un modelo de Frayer sobre gráficos de barras antes de crear un gráfico de barras escalado utilizando los datos recopilados durante el juego.

\_Estándares abordados: \_

- **CCSS.ELA-LITERACY.RF.3.4:** leer con suficiente precisión y fluidez para apoyar la comprensión.
- **CCSS.ELA-LITERACY.SL.3.4:** informar sobre un tema o texto, contar una historia o relatar una experiencia con hechos apropiados y detalles descriptivos relevantes, hablando claramente a un ritmo comprensible.
- **CCSS.MATH.CONTENT.3.MD.B.3:** dibujar un gráfico de imágenes a escala y un gráfico de barras a escala para representar un conjunto de datos con varias categorías. Resolver problemas de uno y dos pasos de "cuántos más" y "cuántos menos" utilizando la información presentada en gráficos de barras escaladas.
- **NGSS.3-PS2-2:** realizar observaciones y/o mediciones del movimiento de un objeto para proporcionar evidencia de que un patrón se puede utilizar para predecir el movimiento futuro.



Esta obra está disponible bajo una **Licencia Creative Commons (CC BY-NC-SA 4.0)**.

**Contáctanos** si desea contar con la licencia de los materiales de Code.org para uso comercial.

# Lección 7: Construye un juego de la Guerra de las Galaxias

60 minutos

## Resumen

En esta lección de **construcción de destechas**, los estudiantes practicarán el uso de eventos para construir un juego que puedan compartir.

## Propósito

Los fundamentos de las ciencias de la computación no buscan solamente enseñar computación, sino también cómo hacer que sea divertida y emocionante. En estas actividades, los estudiantes aprenderán sobre los eventos acompañados por conocidos personajes de *La guerra de las galaxias*. Estos desafíos rompen la distancia entre “aprender” y “divertirse”. Además, los estudiantes aprenderán a reconocer prácticas comunes de programación en juegos, de manera que cuando jueguen en casa, puedan identificar principios básicos de computación en acción.

## Estándares

Curso Completo Alineamiento

### Estándares de Ciencias de la Computación CSTA K-12 (2017)

- ▶ **AP** - Algorithms & Programming

## Oportunidades multidisciplinarias

### Estándares Científicos de la Próxima Generación

- ▶ **ETS** - Engineering in the Sciences

## Agenda

### Actividad previa (15 minutos)

Introducción

### Actividad Principal (30 minutos)

Construye un juego de la Guerra de las Galaxias

### Actividad de cierre (15 minutos)

Reflexión

## Objetivos

Los estudiantes podrán:

- Crear un juego animado e interactivo usando secuencias y eventos.
- Identificar acciones que se correlacionan con eventos de entrada.

## Preparación

- Realiza los desafíos para encontrar cualquier área potencialmente problemática para tu clase.
- Asegúrate de que cada estudiante tenga su Diario de apuntes.

## Enlaces

**¡Aviso!** Por favor, haga una copia de cualquier documento que planee compartir con los estudiantes.

Para los profesores

- **Pauta con ejemplos resueltos** - Guía de respuestas

▼ Hacer una copia

Para los estudiantes

- **Momento de reflexión en línea** - Vídeo

## Vocabulario

- **Evento** - una acción que hace que algo suceda.

## Guía Didáctica

### Actividad previa (15 minutos)

#### Introducción

En una conversación con la clase, pregúntales cuáles son sus videojuegos favoritos (quizás sea necesario recordarles que sólo deben mencionar juegos apropiados para su edad). Pregúntales cuáles son sus partes favoritas del videojuego.

En la mayoría de los casos, los estudiantes responderán con alguna clase de evento. Cuando logres reconocer una respuesta de un estudiante que describa un evento, pídele que continúe con su explicación.

Una vez que el estudiante termine de describir eso que le divierte, dedica algo de tiempo para relacionarlo con la definición de **evento**.

- **Evento**: una acción que causa que algo pase.

Pide a los estudiantes que intenten relacionar algunas de sus partes favoritas de los videojuegos con la descripción de evento. Pídeles que compartan en parejas y que discutan las diferencias entre sus eventos y los de sus compañeros.

#### Consejo didáctico

Si no estás seguro si las respuestas de los estudiantes describen eventos, intenta dividir las respuestas. ¿Hay una acción y una respuesta?

Por ejemplo:

- Cruzar la línea de meta y que los personajes te feliciten en la pantalla.
- Encontrar un gran cofre del tesoro (u otro objeto) y ver cómo crece tu inventario.
- Comprar objetos nuevos en la tienda del juego y usar dicho objeto.
- Presionar un botón del control y que el personaje haga algo genial.

### Actividad Principal (30 minutos)

#### Construye un juego de la Guerra de las Galaxias

##### Desafíos en línea

Es muy probable que los estudiantes estén muy emocionados por desarrollar sus propios juegos de *La guerra de las galaxias* al final de esta serie de desafíos. Si hay tiempo, pídeles que planifiquen qué quieren que haga el juego. La planificación y preparación ayudará a los estudiantes a reconocer de mejor manera los conceptos clave que se intentan enseñar en esta lección. Alienta a los estudiantes a compartir y modificar (remix) los juegos de los demás al final de la lección.



1

Vídeo: Proyecto Star Wars



2-9

Desarrollo de Habilidades

2

3

4

5

6

7

8

9



10

Juego libre

💡 Consejo didáctico

Recuerda a los estudiantes que sólo deben compartir su trabajo con sus amigos o familiares cercanos. Para más información, visualiza o muestra a la clase el video:

**\*Momento para reflexionar en línea.**

## Actividad de cierre (15 minutos)

### Reflexión

\*\*Sugerencias: \*\*

- ¿Sobre qué se trataba la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- Da un ejemplo de un evento que hayas usado hoy en tu programa
- ¿Por qué es importante no compartir información privada en internet? ¿Cómo sabes que es información privada?



Esta obra está disponible bajo una [Licencia Creative Commons \(CC BY-NC-SA 4.0\)](https://creativecommons.org/licenses/by-nc-sa/4.0/).

**Contáctanos** si desea contar con la licencia de los materiales de Code.org para uso comercial.

# Lección 8: Fiesta de baile

42 minutos

## Resumen

En esta lección de **desarrollo de estudios**, los estudiantes programarán una fiesta de baile interactiva.

## Propósito

Esta lección presenta los conceptos básicos de las ciencias de la computación para la codificación y la programación de eventos (con bloques).

## Estándares

Curso Completo Alineamiento

### Estándares de Ciencias de la Computación CSTA K-12 (2017)

- ▶ **AP** - Algorithms & Programming

## Agenda

### Primeros pasos (5 minutos)

#### Preparando el terreno

### Actividad (30 minutos)

#### Codifica tu propia fiesta

#### Apoyo: nivel por nivel

### Actividad de cierre (7 minutos)

#### Reflexión

#### Evaluación

#### Oportunidad multidisciplinaria

## Objetivos

Los estudiantes podrán:

- Crear animaciones de baile a través del código
- Desarrollar programas que respondan a eventos cronometrados
- Desarrollar programas que respondan a las entradas de los usuarios

## Preparación

- Realiza los desafíos para encontrar cualquier área potencialmente problemática para tu clase.
- Asegúrate de que cada estudiante tenga su Diario de apuntes.
- Considera el uso de audífonos. Esta actividad se basa en el sonido.

## Enlaces

**¡Aviso!** Por favor, haga una copia de cualquier documento que planea compartir con los estudiantes.

Para los profesores

- **Niveles de programación abiertos** - Guía de respuestas
- **Lista de Reproducción en Spotify (todas las edades)**

Para los estudiantes

- **Guía de trabajo - Proyecto de fiesta de baile** - Guía de trabajo
- **Fiesta de baile: sin conexión**  
- Recurso

## Vocabulario

- **Evento** - una acción que hace que algo suceda.
- **Programa** - un algoritmo que se ha programado como algo que puede ser ejecutado por una máquina.
- **code** - (v) escribir código, o escribir instrucciones para una computadora.

## Guía Didáctica

### Primeros pasos (5 minutos)

#### Preparando el terreno

Dales la bienvenida a los alumnos a la clase y presenta muy brevemente la actividad del día.

#### Consejo didáctico

Si tienes tiempo y quieres preparar a tus estudiantes con una actividad sin conexión, considera la posibilidad de realizar la actividad **\*Fiesta de baile sin conexión** antes de esta lección. Esta breve lección introduce la idea de eventos al producir pasos de baile diferentes.

#### *Observaciones*

Hoy vamos a hacer algo realmente creativo. ¿Cuál es tu forma favorita de ser creativo?

Alienta a los estudiantes a compartir las formas en que expresan su creatividad, como el arte, el baile, la música, la escritura, etc.

Explica que hoy vamos a ser creativos con nuestros códigos. Así como el color que elegimos para pintar o las palabras que decidimos escribir pueden expresar nuestra creatividad, elegir qué código escribir y cómo interactuará con los usuarios, también puede ser una oportunidad para expresar tu creatividad!

**Levántate y baila:** explica a la clase que hoy vamos a ver cómo combinar la codificación y el baile, de una forma muy creativa. Pide a tus estudiantes que hagan el floss (baile de Fortnite), el dab (pose de victoria) o el paso de baile que quieran por 10 segundos, para entrar en ambiente. Puedes poner una canción de esta **\*Lista de reproducción de Spotify** para ponerle ritmo a la actividad y motivar a los estudiantes. Graba un video de los pasos de baile de tu clase.

[[0]]

# Actividad (30 minutos)

Codifica tu propia fiesta

## Filtro de música

Este tutorial utiliza canciones de artistas populares. Para tener una vista previa de las canciones de este tutorial, revisa esta **\*lista de reproducción de Spotify**. Utilizamos versiones seguras para radio para menores de 13 años y limitamos las canciones a esta lista filtrada. Si quieres usar la lista filtrada con los estudiantes más grandes, puedes compartir esta **\*lista de reproducción de Spotify** con tu clase.

Apoyo: nivel por nivel

Nivel 1



**Vídeo: Calentamiento**

Nivel 2

- Arrastra el bloque rojo `crear un nuevo` desde la caja de herramientas, en la izquierda, hasta el espacio de trabajo, en la derecha. Conéctalo dentro del bloque `establecer`.
- Acabas de escribir tu primer programa. Asegúrate de presionar Ejecutar para ver qué pasa. Deberías escuchar música y ver a un personaje comenzar a moverse en el área de visualización.



**Práctica**

Nivel 3



**Vídeo: Eventos**

Nivel 4

- Los niveles 4 y 5 se tratan de hacer que el baile sea interactivo.
- Los bloques verdes son bloques de evento. Estos bloques comienzan una nueva secuencia de código y no necesitan estar conectados dentro del bloque `establecer`.
- Conectar el bloque púrpura bajo el bloque de evento verde te permite hacer que el personaje realice un paso de baile al presionar alguno de los botones naranjas con flechas o las teclas del teclado.
- Asegúrate de presionar Ejecutar antes de presionar las flechas, sino el personaje no hará nada.

Nivel 5

- Asegúrate de arrastrar un segundo bloque púrpura `hacer una vez`. Deberías tener un bloque `osos hacer una vez` y un bloque `gatos hacer una vez` en tu espacio de trabajo. Ambos deben estar conectados a un bloque de evento verde `al pulsar`.
- Asegúrate de presionar Ejecutar antes de presionar las flechas, sino los personajes no harán nada.

 4-5

Eventos

4

5

Nivel 6

 6

Vídeo: Compases

Nivel 7

- Los niveles 7 y 8 se tratan de sincronizar el baile con la música.
- Los bloques de evento `después medidas` también comienzan una nueva secuencia de código y no deben estar conectados dentro del bloque `establecer`.
- Conectar el bloque púrpura `hacer para siempre` debajo del bloque de evento verde `después medidas` debería hacer que el personaje realice un paso de baile después de la cantidad de medidas que indique.
- El bloque `hacer para siempre` funciona de forma diferente al bloque hacer una vez, visto en niveles anteriores.

Nivel 8

- Asegúrate de arrastrar un segundo bloque de evento verde. Debería tener un bloque `después 4 medidas` y un bloque `después 6 medidas` en tu espacio de trabajo. Ambos deben tener un bloque púrpura conectado abajo.

Nivel 9

- El nivel 9 se trata de crear grupos de bailarines rápidamente.
- Usa el nuevo bloque de la caja de herramientas para crear un grupo de pequeños bailarines \*Debes usar, además, el bloque normal `crear un nuevo` para crear un bailarín "líder".
- Muchos estudiantes estarán familiarizados con la idea de hacer que algo parezca estar más lejos al dibujarlo en una escala más pequeña. En el siguiente nivel, podrás afinar este efecto.

 7-9

Compases

7

8

9

Nivel 10

 10

Vídeo: Propiedades

Nivel 11

- Los niveles 11, 12 y 13 se tratan de ajustar las propiedades (tamaño, color) de los bailarines.
- Es importante que te asegures de que el bloque verde `establecer` esté ubicado en algún lugar del programa *después* de que los bailarines hayan sido creados. Para resolver este desafío,

coloca un bloque `establecer tamaño` en cualquier lugar de tu programa y úsalo para cambiar el tamaño de uno de los bailarines.

- Los bailarines creados en grupos tienen un tamaño por defecto de 30. Los demás bailarines tienen un tamaño por defecto de 100.

#### Nivel 12

- Así como en el nivel anterior, asegúrate de usar el bloque `establecer tonalidad a (color)` después de haber creado a los bailarines en el programa. Por ejemplo, si lo primero que hace es colocar este bloque dentro del bloque `establecer`, no tendrá ningún efecto.

#### Nivel 13

- Con el código indicado, deberías ver al bailarín cambiar de colores, tamaños y pasos de baile repetidamente.
- Asegúrate de que haya un bloque verde agua `cambiar`, un bloque verde agua `distribuir aleatoriamente` o un bloque púrpura `hacer para siempre` conectado dentro del bloque `cada 2 medidas`.
- Asegúrate de que los bloques `hacer para siempre` estén configurados como (siguiente), (anterior) o (aleatorio). De lo contrario, el bailarín realizará el movimiento configurado repetidamente.
- Nota que el **código escrito por los estudiantes, en este nivel, no está siendo corregido**. Esto significa que no importa lo que hagan, siempre pasarán este nivel, ya que el objetivo es que se sientan libres de experimentar con sus códigos de las formas que más les llamen la atención. Haz clic en "Finalizar" para continuar.
- Crear nuevos bailarines dentro del bloque `cada 2 medidas` provocará que el programa cree copias idénticas del bailarín en la misma ubicación. ¡Esto podría llevar a consecuencias indeseadas!

 11-13

Propiedades

11

12

13

 Consejo didáctico ▲

A estas alturas de la lección puedes observar que las instrucciones son menos exigentes. Anima a los alumnos a ser creativos y a explorar los nuevos bloques. A partir de este momento, el código de los alumnos **no se corrige** para, así, fomentar la experimentación en lugar de resolver una tarea específica.

#### Nivel 14

 14

Vídeo: Recapitulación

#### Nivel 15

- Este último nivel es abierto. El tutorial mismo está diseñado para dar a los estudiantes mucho tiempo para trabajar en sus propios bailes.
- **Alienta a compartir:** si los estudiantes tienen celulares y cuentan con plan de datos, pueden fácil y rápidamente enviar un enlace de sus proyectos a sus celulares o al de un amigo. Aliéntalos a hacerlo siempre y cuando las políticas del colegio lo permitan.

- **\*\* Alienta la creatividad:\*\*** la creatividad ha sido importante a lo largo de esta lección; sin embargo, ¡aquí es donde podrás brillar!

 15

Juego libre

 16-17

Lecciones adicionales



## Actividad de cierre (7 minutos)

### Reflexión (5 minutos)

- Plantea una pregunta que tenga múltiples respuestas, como "¿Qué es lo que te ha gustado de la actividad de hoy?" o "¿Qué relación hay entre la creatividad y la informática?".

### Evaluación (2 minutos)

Pídele a los alumnos que añadan sus notas adhesivas "importantes" o tarjetas de notas a su mapa mental de "Ciencias de Computación" antes de que salgan de clase. Trata de llenar el cuadro con muchas ideas acerca de lo que son las Ciencias de Computación y por qué son importantes.

### Oportunidad multidisciplinaria

### Supervivencia de los bailarines (45-60 minutos)

#### Ciencias de la computación + Ciencia

**Supervivencia de los bailarines** es una actividad opcional alineada con las normas de ciencia de próxima generación, escrita por nuestra comunidad docente. Con la fiesta de baile, los estudiantes seleccionarán al menos un organismo (bailarín) que sea adecuado para el entorno (mundo) que crean. Se anima a los estudiantes a utilizar los bloques de código "propiedades" para personalizar las características de cada organismo (bailarín).

*Estándares abordados:*

- **NGSS.3-LS4-2:** utilizar evidencia para elaborar una explicación de cómo las variaciones en las características entre individuos de la misma especie pueden proporcionar ventajas para sobrevivir, encontrar parejas y reproducirse.
- **NGSS.3-LS4-3:** construir un argumento con evidencia de que en un hábitat determinado algunos organismos pueden sobrevivir bien, otros sobreviven menos bien y otros no pueden sobrevivir en absoluto.
- **EXTENSIÓN OPCIONAL: NGSS.3-LS4-4:** hacer una afirmación sobre el mérito de una solución a un problema causado cuando cambia el entorno y los tipos de plantas y animales que viven allí pueden cambiar.



Esta obra está disponible bajo una **Licencia Creative Commons (CC BY-NC-SA 4.0)**.

**Contáctanos** si desea contar con la licencia de los materiales de Code.org para uso comercial.

# Lección 9: Bucles de la Era de Hielo

45 minutos

## Resumen

Esta lección de **contexto-configuración/ desarrollo de calificaciones** introducirá rápidamente a los estudiantes a los bucles.

## Propósito

En esta lección, los estudiantes aprenderán más sobre los bucles y cómo implementarlos en un código Blockly. El uso de bucles es una habilidad importante en programación, ya que repetir comandos manualmente es tedioso e ineficiente. A través de estos desafíos de Code.org, los estudiantes aprenderán a añadir instrucciones a bucles ya existentes, agrupar en bucles códigos repetidos y reconocer patrones que necesitan ser repetidos.

## Estándares

Curso Completo Alineamiento

### Estándares de Ciencias de la Computación CSTA K-12 (2017)

► **AP** - Algorithms & Programming

## Agenda

### Actividad previa (10 minutos)

#### Introducción a los bucles

### Actividad Principal (30 minutos)

#### Bucles de la Era de Hielo

### Actividad de cierre (5 minutos)

#### Reflexión

### Ampliación del aprendizaje

## Objetivos

Los estudiantes podrán:

- Construir un programa usando estructuras que repitan áreas de código
- Mejorar un código existente al encontrar áreas de repetición y transformarlas en estructuras de bucle

## Preparación

- (Opcional) Elige un par de desafíos para hacer en conjunto con la clase.
- Asegúrate de que cada estudiante tenga su Diario de apuntes.

## Enlaces

**¡Aviso!** Por favor, haga una copia de cualquier documento que planea compartir con los estudiantes.

Para los estudiantes

- **Caras de emociones** - Imágenes de emociones
- **Programación en parejas** - Video del estudiante
- **Momento de reflexión en línea** - Video

## Vocabulario

- **Bucle** - la acción de hacer algo una y otra vez.

- **Repetir** - Hacer algo otra vez.

## Guía Didáctica

### Actividad previa (10 minutos)

#### Introducción a los bucles

**Modelar:** elige a un voluntario y que se ponga de pie.

- Indica al voluntario que dé una vuelta alrededor de la mesa (o de una silla, o de un amigo).
- Cuando termine, indícale que lo haga nuevamente, usando exactamente las mismas palabras usadas anteriormente.
- Cuando termine, indícale que lo haga nuevamente.
- Y otra vez.

**Preguntar:** *¿No hubiese sido más fácil para mí pedirte que dieras cuatro vueltas alrededor de la mesa?*

**Reflexión:** *¿Qué tal si hubiese querido que lo hicieras 10 veces? ¿Cómo podría haber hecho que mis instrucciones fuesen más eficientes sin tener que repetirlas tantas veces? Siéntanse libres de anotar sus instrucciones en una hoja.*

**Exposición:** pídeles a unos cuantos estudiantes que compartan sus instrucciones con la clase. Remarca que cada aporte simplifica el enfoque general para dar instrucciones.

**Di:** *Hoy vamos a trabajar en encontrar formas para facilitar el dar muchas instrucciones, especialmente cuando esas instrucciones se repiten mucho.*

#### Revisión de bucles de La era de hielo

Para terminar la conexión, revisen un desafío en línea (o dos) en conjunto como clase.

**Modelar:** muestra un desafío en línea completo de la lección. Recomendamos el desafío número 5. Muestra el “área de juego”, en donde está *Scrat* y la bellota, y el “espacio de trabajo”, donde están los bloques Blockly. Explica que, ahora, este código Blockly es el lenguaje que usará la clase para ayudar a *Scrat* a conseguir la bellota. ¿Los estudiantes pueden ver alguna similitud con el ejercicio que acaban de hacer? ¿Cuáles son las diferencias más grandes?

Trabaja con tu clase para arrastrar códigos en el espacio de trabajo, de manera que *Scrat* pueda (finalmente) conseguir la bellota.

**Transición:** los estudiantes deberían estar listos para pasar a los computadores y completar los desafíos por ellos mismos.

### Actividad Principal (30 minutos)

#### Bucles de la Era de Hielo

Mientras los estudiantes realizan los desafíos, mira si pueden darse cuenta de la diferencia en la cantidad de bloques que utilizan cuando usan bucles y la cantidad de bloques que utilizan cuando no los usan.

1-2

Desarrollo de Habilidades

1

2

3

Vídeo: Utilizando el Bloque Repetir

4-9

Desarrollo de Habilidades

4

5

6

7

8

9

10

Desafío

11-12

Práctica

11

12

Consejo didáctico

Enseña a los estudiantes la manera *correcta* de ayudar a sus compañeros:

- No sentarse en la silla del compañero.
- No usar el teclado del compañero.
- No tocar el mouse del compañero.
- Asegurarte de que tu compañero pueda describirte la solución en voz alta antes de que te vayas.

**Observación:** los docentes son de vital importancia en la educación de las ciencias de la computación y juegan un rol fundamental para producir un ambiente vibrante y colaborativo en la sala de clases. Durante las actividades en línea, el rol del docente es alentar y apoyar. Los desafíos en línea están estructurados para estar centrados en el estudiante, así que los docentes deben evitar involucrarse cuando los estudiantes tengan problemas para resolverlos. Algunas ideas de cómo hacerlo son:

- Usa el video **\*Momento para reflexionar en línea** cada vez que sea posible durante la actividad.
- A través de desafíos o preguntas, anima a los estudiantes para que busquen respuestas con sus respectivas parejas.
- Las preguntas sin responder pueden ser delegadas a un grupo cercano, que podría ya tener la respuesta.
- Recuérdales usar el proceso de depuración antes de que te acerques a ayudar.
- Pide a los estudiantes que describan el problema que estén viendo. ¿Qué se supone que debe hacer?, ¿qué hace?, ¿qué te dice eso?
- Recuérdales que la frustración es un paso en el camino del aprendizaje y que la perseverancia dará sus frutos.

- Si un estudiante sigue con problemas para avanzar después de las acciones anteriores, haz preguntas clave para que los estudiantes identifiquen el error por ellos mismos.

## Actividad de cierre (5 minutos)

### Reflexión

#### Sugerencias:

- ¿Sobre qué se trataba la lección de hoy?
- Dibuja tu propio laberinto con *Scratch* intentando conseguir la bellota. ¿Los bucles te ayudarán a resolverlo?
- Dibújate a ti mismo usando un bucle para realizar una actividad diaria, como lavarte los dientes.

## Ampliación del aprendizaje

Utiliza estas actividades para mejorar el aprendizaje de los estudiantes. Se pueden utilizar como actividades extraprogramáticas.

#### A moverse

- Dale a los estudiantes imágenes de acciones o pasos de baile que puedan hacer.
  - Pídeles que realicen los pasos de baile y añadan bucles para crear sus propias coreografías.
- Comparte los bailes con el resto de la clase.

#### Vuelve a conectarlo

- Busca algunos videos en YouTube de bailes populares que sean repetitivos.
- ¿Los estudiantes pueden descifrar el bucle?
- ¡Intenta lo mismo con canciones!



Esta obra está disponible bajo una [Licencia Creative Commons \(CC BY-NC-SA 4.0\)](https://creativecommons.org/licenses/by-nc-sa/4.0/).

**Contáctanos** si desea contar con la licencia de los materiales de Code.org para uso comercial.

# Lección 10: Dibujar formas con bucles

60 minutos

## Resumen

Esta lección, basada en la comprensión de los bucles de lecciones anteriores, brinda a los estudiantes la oportunidad de ser verdaderamente creativos. Esta actividad también sirve como un ejercicio de depuración para seguir practicando la resolución de problemas.

## Propósito

En esta lección se hace énfasis en el poder de los bucles, ya que está destinada a suscitar una reflexión sobre la superioridad de los bucles `repetir` frente a la escritura manual.

## Estándares

Curso Completo Alineamiento

### Estándares de Ciencias de la Computación CSTA K-12 (2017)

- ▶ **AP** - Algorithms & Programming

## Agenda

### Actividad previa (15 minutos)

#### Introducción

### Actividad Principal (30 minutos)

#### Dibujar formas con bucles

### Actividad de cierre (15 minutos)

#### Reflexión

## Objetivos

Los estudiantes podrán:

- Diferenciar entre comandos que deben ser repetidos en bucle y comandos que deben ser usados de forma casual.
- Identificar los beneficios de usar una estructura de bucle en lugar de la repetición manual.

## Preparación

- Juega con los rompecabezas para encontrar cualquier área potencialmente problemática para tu clase.
- Asegúrate de que cada estudiante tenga un diario de reflexión.

## Vocabulario

- **Bucle** - la acción de hacer algo una y otra vez.
- **Repetir** - Hacer algo otra vez.

## Guía Didáctica

### Actividad previa (15 minutos)

#### Introducción

En este punto, los estudiantes ya deben conocer los bucles. En base a lo que consideres que podría beneficiar más a tu clase, recomendamos:

- Repasar un desafío de la lección anterior
- Repasar un desafío de esta lección

¡Todas estas opciones te ayudarán a preparar a tu clase para divertirse con los desafíos en línea!

## Actividad Principal (30 minutos)

Dibujar formas con bucles

 1 Presentación del Artista con JR Hildebrand

 2-3 Desarrollo de Habilidades

2 3

 4 Vídeo: Bucles con el Artista

 5-9 Desarrollo de Habilidades

5 6 7 8 9

 10 Desafío

 11 Práctica

### Desafíos en línea

Para algunos estudiantes, identificar en dónde agregar bucles `repetir` podría ser más fácil si escriben primero el programa sin bucles y luego encierran en un círculo las secciones que se repitan. De ser así, podría ser de gran ayuda para ellos permitirles tener hojas y lápices a mano. Los estudiantes también podrían disfrutar dibujar algunas formas y figuras en una hoja antes de programarlas en digital.

## Actividad de cierre (15 minutos)

### Reflexión

#### Sugerencias:

- ¿Sobre qué se trataba la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- ¿Cuál fue la figura o forma más genial que programaste hoy? ¡Dibújala!
- ¿Qué otra figura o forma podrías programar con bucles? ¿Puedes inventar el código para crearla?



Esta obra está disponible bajo una [Licencia Creative Commons \(CC BY-NC-SA 4.0\)](https://creativecommons.org/licenses/by-nc-sa/4.0/).

**Contáctanos** si desea contar con la licencia de los materiales de Code.org para uso comercial.

# Lección 11: Bucles anidados en Laberinto

55 minutos

## Resumen

En esta lección de **desarrollo de habilidades**, los estudiantes aprenderán a programar un bucle dentro de otro bucle.

## Propósito

En esta introducción a los bucles anidados, los estudiantes tendrán que salir de su zona de confort para idear soluciones más eficientes.

En desafíos anteriores, los estudiantes fueron guiados para identificar patrones repetitivos. Ahora, los estudiantes aprenderán a identificar patrones *dentro* de patrones repetidos para desarrollar bucles anidados. Esta lección comienza animando a los estudiantes a intentar resolver un desafío cuyo código es irritante, complejo y extenso de escribir. Luego de ver un video introductorio sobre bucles anidados, se les mostrará un ejemplo y se les pedirá predecir qué pasa si se pone un bucle dentro de otro bucle. Este sistema progresivo les permite tener mucha práctica a fin de fortalecer y desarrollar su comprensión de los bucles.

## Estándares

Curso Completo Alineamiento

### Estándares de Ciencias de la Computación CSTA K-12 (2017)

► **AP** - Algorithms & Programming

## Agenda

### Actividad previa (10 minutos)

#### Introducción

### Actividad Principal (30 minutos)

#### Bucles anidados en Laberinto

### Actividad de cierre (15 minutos)

#### Reflexión

## Objetivos

Los estudiantes podrán:

- Dividir tareas complejas en secciones repetibles más pequeñas.
- Identificar los beneficios de usar una estructura de bucle en lugar de la repetición manual.
- Reconocer que largos patrones repetidos están conformados de patrones repetidos más pequeños.

## Preparación

- Juega con los rompecabezas para encontrar cualquier área potencialmente problemática para tu clase.
- Asegúrate de que cada estudiante tenga un diario de reflexión.

## Vocabulario

- **Comando** - Una instrucción para un computador. Muchos comandos juntos forman algoritmos y programas.
- **Bucle** - la acción de hacer algo una y otra vez.
- **Repetir** - Hacer algo otra vez.

# Guía Didáctica

## Actividad previa (10 minutos)

### Introducción

Repasa brevemente con tu clase qué son y para qué sirven los **bucles**.

- \_ ¿Qué hacen los bucles?\_
  - Los bucles repiten una serie de comandos.
- \_ ¿Para qué usamos los bucles?\_
  - Usamos los bucles para crear patrones de acciones repetitivas.

Coméntales que ahora harán algo súper genial: usar bucles dentro de otros bucles. Pregúntales si tienen alguna idea de para qué podríamos usar un bucle dentro de otro bucle.

*Si un bucle repite un patrón, ¡entonces un bucle dentro de otro bucle repetiría un patrón de patrones!*

Los estudiantes no necesitan entender esto de inmediato, así que siéntete libre de pasar a los desafíos en línea incluso si los estudiantes todavía parecen un poco confundidos.

## Actividad Principal (30 minutos)

### Bucles anidados en Laberinto

Para esta lección, recomendamos encarecidamente usar la programación en parejas, ya que este tema podría ser complicado para la mayoría de tus estudiantes. Trabajar con un compañero y discutir posibles soluciones para los desafíos podría aliviar tu carga mental.

Recomendamos además tener papel y lápices a mano para que los estudiantes puedan planificar antes de codificar. Algunos desafíos tienen bloques con límite de usos, por lo que los estudiantes podrían necesitar escribir la versión larga del código para analizarlo y encontrar las zonas de repetición.



1-2

Desarrollo de Habilidades

1

2



3

Vídeo: Bucles Anidados con la Abeja



4

Predicción



5-9

Desarrollo de Habilidades

5

6

7

8

9



Desafío



Práctica



Predicción



Lecciones adicionales



## Actividad de cierre (15 minutos)

### Reflexión

#### Sugerencias:

- ¿Sobre qué se trataba la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- ¿Qué es un bucle anidado?
- ¿Puedes dibujar un desafío que use un bucle anidado? intenta codificar la solución de tu propio desafío.



Esta obra está disponible bajo una **Licencia Creative Commons (CC BY-NC-SA 4.0)**.

**Contáctanos** si desea contar con la licencia de los materiales de Code.org para uso comercial.

# Lección 12: Condicionales con tarjetas

55 minutos

## Resumen

En esta lección de **contexto**, los estudiantes escribirán *declaraciones condicionales (si/si no)* para establecer las reglas de los juegos de barajas simples.

## Propósito

Lo mejor de enseñar condicionales es que los estudiantes ya conocen el concepto, ya que lo ven a diario en sus vidas.

Esta lección incorpora las ciencias de la computación en el mundo real, desarrollando la capacidad de notar si una condición es verdadera o falsa. Los estudiantes aprenderán a usar la declaración *si* para establecer cuándo debe ejecutarse un comando en específico. También aprenderán a usar la declaración *si / si no* para establecer cuándo debe ejecutarse un comando y qué hacer si no es ejecutado. Es posible que los estudiantes no reconozcan el término condicional, pero la mayoría estarán familiarizados con la idea de anteponer el término "si" en una oración para asegurarse de que una acción ocurra sólo bajo ciertas circunstancias.

## Estándares

Curso Completo Alineamiento

### Estándares de Ciencias de la Computación CSTA K-12 (2017)

- ▶ **AP** - Algorithms & Programming

## Agenda

### Actividad inicial (15 minutos)

#### Introducción

### Actividad Principal (25 minutos)

#### Condicionales con tarjetas

### Actividad de cierre (10 minutos)

#### Reflexión

#### Reflexión

### Evaluación (5 minutos)

### Ampliación del aprendizaje

### Oportunidad multidisciplinaria

## Objetivos

Los estudiantes podrán:

- Definir circunstancias en que ciertas partes de un programa deban ejecutarse y cuándo no.
- Determinar cuándo se cumple una condicional, en base a ciertas normas.
- Seguir un programa y predecir el resultado, dada una serie de entradas.

## Preparación

- Prepara barajas de cartas o algo similar.
- Imprime o proyecta los **\*programas de ejemplo**.
- Imprime una \ **\*evaluación** para cada estudiante.
- Asegúrate de que cada estudiante tenga su Diario de apuntes.

## Enlaces

**¡Aviso!** Por favor, haga una copia de cualquier documento que planea compartir con los estudiantes.

Para los profesores

- **Condicionales con tarjetas** - Guía de respuestas de la evaluación  
[▼ Hacer una copia](#)
- **Condicionales con tarjetas** - Video de la evaluación

- **Condicionales con tarjetas** - Video - Lección en acción
- **Programa ejemplo - Condicionales con cartas** - Guía de preparación para el docente

Para los estudiantes

- **Condicionales con tarjetas** - Evaluación
- **Condicionales con tarjetas** - Video sin conexión (**Descarga**)

## Vocabulario

- **Condicionales** - sentencias que solo se ejecutan bajo ciertas condiciones.
- **seudocódigo** - Descripción en lenguaje sencillo de los pasos de un algoritmo

## Guía Didáctica

### Actividad inicial (15 minutos)

#### Vocabulario

Esta lección introduce una nueva e importante palabra:

**Condicional:** *díganlo conmigo, CON - DI - CIO - NAL.*

Declaraciones que sólo se ejecutan bajo ciertas circunstancias.

#### Introducción

- Podemos comenzar esta lección de inmediato.
  - Coméntale a la clase que si pueden mantenerse completamente en silencio por 30 segundos, vas a:
    - Cantar una ópera.
    - Dar 5 minutos más de recreo.
    - Pararte de manos.
  - Comienza a contar.
  - Si los estudiantes lo logran, indícales que lo lograron y dales su recompensa.
  - De lo contrario, indícales que no lograron mantenerse completamente en silencio por 30 segundos, por lo que no tendrán la recompensa.
- Pregúntales cuál era la condición para obtener la recompensa.
  - La condición era SI estaban en silencio por 30 segundos.

- De ser así, la condición sería verdadera y obtendrían la recompensa.
- De no ser así, la condición sería falsa y no obtendrían la recompensa.
- *¿Se les ocurre otra condicional?*
  - Si pueden adivinar mi edad, toda la clase les dará un aplauso.
  - Si sé la respuesta, puedo levantar la mano.
  - ¿Qué otro ejemplo se les ocurre?
- *A veces, queremos tener una condición adicional, en caso de que la declaración "SI" sea falsa.*
  - *\_ A esta condición adicional se le llama declaración "SI NO".\_*
  - *Si la declaración "SI" no se cumple, podemos ver la declaración "SI NO" para saber qué hacer.*
    - *Ejemplo: SI saco un rey de esta baraja de cartas, todos aplauden. SI NO, todos dirán "OOOUUU".*
    - *\_ Intentémoslo\_ (saca una carta y mira si tu clase reacciona de forma apropiada)*
  - *Pide a la clase analizar lo que acaba de pasar.*
    - *¿Cuál era la condición SI?*
    - *¿Cuál era la condición SI NO?*
    - *¿Qué condición se cumplió?*
  - *Créanlo o no, aún tenemos una opción más.*
    - *Qué tal si quiero que aplaudan cuando saque un 7, si no, si es menor a 7 ustedes dirán "¡SI!", si no ustedes dirán "OOOUUU".*
      - *Para eso tenemos los términos condicional SI, condicional SI NO-SI, y condicional SI NO.*
      - *SI es la primera condición.*
      - *SI NO-SI se cumple sólo si la condición SI no es verdadera.*
      - *SI NO se cumple sólo si ninguna de las condiciones anteriores es verdadera.*

Ahora vamos a jugar un juego.

## Actividad Principal (25 minutos)

### Condicionales con tarjetas

Actividad sin cone...

Condicionales con Cartas - Actividad Desenchufada

### Condicionales con tarjetas: programa ejemplo - Guía de preparación del docente

#### Instrucciones:

- Con tu clase, desarrolla unos cuantos programas que dependan de cosas como el diseño de la carta, el color o el valor, para ganar o perder puntos. Puedes escribir los programas como algoritmos, pseudocódigo o código.

Aquí hay un ejemplo de un programa escrito con un algoritmo:

```

si (CARTA es ROJA)
    Tu equipo gana 1 punto

si no
    El otro equipo gana 1 punto

```

Aquí hay un ejemplo del mismo programa escrito con pseudocódigo:

```
si (color.carta == ROJO){
    puntos.tuyos = puntos.tuyos + 1;
}

si no {
    puntos.otro = puntos.otro + 1;
}
```

- Divide tu clase en equipos.
- Cada equipo debe tener una pila de barajas cerca (mínimo tantas barajas como miembros haya en el equipo).
- Coloca uno de los “programas” en la pizarra para que todos lo puedan ver.
- Haz que los estudiantes se turnen para sacar barajas, siguiendo el programa para ver cuántos puntos ganan en cada ronda.
- Juega varias veces con varios programas diferentes, así ayudarás a los estudiantes a entender bien cómo funcionan las condicionales.

Una vez que la clase haya practicado lo suficiente, puedes alentar a los estudiantes a anidar una condicional dentro de otra. Asegúrate de que entienden que si la carta es roja, TU equipo gana 1 punto y luego *no pasa nada más*, ya que la condición se cumplió:

```
si (CARTA es ROJA)
    TU equipo gana 1 punto

si no
    si (CARTA es mayor a 9)
        El OTRO equipo gana 1 punto
    si no
        TU equipo gana tantos puntos como el número de la carta
```

Aquí está el mismo programa pero en pseudocódigo:

```
si (color.carta == ROJO ){ puntos.tuyos = puntos.tuyos + 1; } si no { si (valor.carta > 9){ puntos.otros = puntos.otros + 1; } si no { puntos.tuyos = puntos.tuyos + valor.carta; } }
```

## Actividad de cierre (10 minutos)

### Reflexión

💡 Consejo didáctico ▲

Las preguntas de la charla rápida están ideadas para suscitar un pensamiento global que relacione la lección con el mundo y el futuro de los estudiantes. Respecto a tu clase, decide si quieres discutir las como clase, en grupos o con sus compañeros de puesto.

Charla rápida: ¿qué aprendimos?

- Si fueses a codificar esto en código Blockly, ¿qué deberías agregarle a la condicional para que el código se ejecutara más de una vez? (un bucle).
- ¿Qué otras cosas haces en tu vida diaria bajo ciertas condiciones?
- Imaginemos que debas hacer algo cuando el valor de una carta sea mayor a 5. Si sacas un 5, ¿cumples esa condición?
- Noten que las condiciones son “verdaderas” o “falsas”. No hay ningún parámetro para una condición que evalúe, por ejemplo, “plátano”.
- Cuando necesitan cumplir con varias condiciones a la vez, podemos usar “condicionales anidadas”.
  - ¿Qué creen que significa eso?
  - ¿Pueden dar un ejemplo de dónde vemos eso en un juego?
- ¿Qué parte de ese juego te gustó más?

## Reflexión

### Sugerencias:

- ¿Sobre qué se trataba la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- ¿Qué es una condicional? ¿Cómo usaste una condicional hoy?
- Nombra algunas condicionales que hayas usado hoy. ¿Se te ocurren otras condicionales que puedas usar con una baraja de cartas?

## Evaluación (5 minutos)

Distribuye la **\*Evaluación – Condicionales con cartas** y permite que los estudiantes realicen la actividad de forma individual, luego de haber explicado claramente las instrucciones. Gracias a la actividad previa, esto no debería significar ningún problema para ellos. Puedes ver el **\*Video de la evaluación** como guía.

## Ampliación del aprendizaje

Usa estas actividades para ampliar el aprendizaje de los estudiantes. Se pueden usar como actividades extras fuera del aula.

### Verdadero/Falso

- Forma a los estudiantes como si fuesen a jugar el juego del semáforo.
- Escoge a un estudiante para que pase al frente como el locutor.
- El locutor elige una condición y les pide a todos que cumplan esa condición para poder dar un paso al frente.
- Si tienes un cinturón rojo, da un paso al frente.
- Si estás usando zapatillas, da un paso al frente.
- Intenten cambiar un poco las cosas, diciendo cosas como “si no eres rubio, da un paso al frente”.

### Anidando

- Divide a los estudiantes en grupos o en parejas.
- Pídeles que escriban declaraciones “si” para cartas en tiritas de papel, tales como: - El diseño es diamantes. - El color es rojo.
- Pídeles que hagan tiritas similares para los resultados. - Gana un punto. - Pierde un punto.
- Una vez que todo esté listo, pídeles que elijan tres tiritas de cada tipo y tres cartas, prestando atención al orden seleccionado.

- En tres hojas de papel, que los estudiantes escriban tres programas diferentes usando sólo las condicionales de las tiritas de papel que eligieron, en cualquier orden.
  - Anima a los estudiantes a poner declaraciones “si” dentro de otras declaraciones “si”.
- Ahora los estudiantes deben ejecutar los tres programas utilizando las cartas que sacaron, en el mismo orden para cada programa. - ¿Algunos programas dieron la misma respuesta? - ¿Alguno arrojó algo diferente?

## Oportunidad multidisciplinaria

Juega a las cartas con condicionales (20-30 minutos)

 **Ciencias de la computación + Lengua y Literatura + Matemáticas**

**Juega a las cartas con condicionales** es una actividad opcional alineada con los estándares de Lengua y Literatura, y Matemáticas de Common Core, escrita por nuestra comunidad docente. Los estudiantes usarán una baraja de cartas para revisar problemas básicos de multiplicación y división a medida que apliques el concepto de condicionales en ciencias de la computación.

*Estándares abordados:*

- **CCSS.ELA-LITERACY.SL.3.3:** formular y responder preguntas sobre la información de un orador, ofreciendo la elaboración y los detalles adecuados.
- **CCSS.MATH.CONTENT.3.OA.C.7:** multiplicar y dividir fluidamente dentro de 100, usando estrategias como la relación entre multiplicación y división (por ejemplo, sabiendo que  $8 \times 5 = 40$ , uno conoce  $40 \div 5 = 8$ ) o las propiedades de las operaciones. Al final del tercer grado, conocer de memoria todos los productos de dos números de un dígito.



Esta obra está disponible bajo una **Licencia Creative Commons (CC BY-NC-SA 4.0)**.

**Contáctanos** si desea contar con la licencia de los materiales de Code.org para uso comercial.

# Lección 13: Mirando hacia el Futuro con Minecraft

50 minutos

## Resumen

En esta lección de **desarrollo de habilidades**, los estudiantes usarán los conceptos que aprendieron en previas lecciones y comenzarán a usar **instrucciones condicionales**!

## Propósito

Este conjunto de desafíos trabajarás para solidificar y desarrollar el conocimiento de los bucles y presentar las condicionales. Al combinar estos dos conceptos, los estudiantes podrán explorar el potencial de crear programas divertidos e innovadores en un entorno nuevo y emocionante.

## Estándares

Curso Completo Alineamiento

### Estándares de Ciencias de la Computación CSTA K-12 (2017)

- ▶ **AP** - Algorithms & Programming

## Agenda

### Actividad previa (15 minutos)

#### Introducción

### Actividad Principal (30 minutos)

#### Mirando hacia el Futuro con Minecraft

### Actividad de cierre (5 minutos)

#### Reflexión

### Ampliación del aprendizaje

## Objetivos

Los estudiantes podrán:

- Definir circunstancias en que ciertas partes de un programa deban ejecutarse y cuándo no.
- Determinar cuándo se cumple una condicional, en base a ciertas normas.

## Preparación

- Realiza los desafíos para encontrar cualquier área potencialmente problemática para tu clase.
- Asegúrate de que cada estudiante tenga su Diario de apuntes.

## Vocabulario

- **Condición** - algo que un programa verifica para ver si es verdadero antes de permitir una acción.
- **Condicionales** - sentencias que solo se ejecutan bajo ciertas condiciones.

# Guía Didáctica

## Actividad previa (15 minutos)

### Introducción

Reúne a la clase y pide a dos voluntarios que caminen en línea recta en alguna dirección en la sala de clase. Si encuentran una silla fuera de lugar, deben pasarla por encima. Si llegan a una pared, deben sentarse.

Una vez que todos los estudiantes estén sentados, pregunta cómo programarían un robot para que responda ante una pared o a una silla. Recuérdales que no pueden decir simplemente "Pasar por encima de la silla" a menos que sepan que hay una silla y que no siempre sabrán que hay una silla. Podría ser útil traducir la tarea en instrucciones como:

- mientras hay un camino por delante
  - camina hacia adelante
  - si hay una silla, pasa sobre ella
- siéntate

Diles a los estudiantes que usarán condicionales durante esta lección. Recuerda la definición de:

- **Condición:** una declaración que un programa comprueba si es verdadera o falsa. Si es verdadera, se lleva a cabo una acción. De lo contrario, se ignora la acción.
- **Condicionales:** declaraciones que sólo se ejecutan bajo ciertas condiciones.

Conversen sobre cuándo pueden usar una condicional en el código.

## Actividad Principal (30 minutos)

### Mirando hacia el Futuro con Minecraft

#### Desafío en línea

A los estudiantes les va a encantar esta lección. Es probable que la mayoría de tus estudiantes hayan oído hablar de *Minecraft*, pero igualmente realiza una breve introducción para aquellos que quizás no lo conozcan.

*Minecraft* es un juego de cubos. Pueden jugar como Alex o Steve mientras recorren laberintos. Tendrán que evitar la lava, recoger objetos y explorar un mundo formado por cubos.

Muestra uno de los desafíos a la clase (te recomendamos el desafío 11). Una vez que hayan respondido todas las preguntas, lleva a los estudiantes a los computadores para que comiencen a programar en parejas.

 1-10

Desarrollo de Habilidades

1

2

3

4

5

6

7

8

9

10

 11-13

Desafío

## Actividad de cierre (5 minutos)

### Reflexión

**\*\*Sugerencia: \*\***

- Dibuja una carita para mostrar cómo te sentiste durante la lección de hoy.
- Dibuja algo más que podrías haber construido en este mundo de *Minecraft*.
- ¿Puedes dibujar una escena en la que alguien esté usando una condicional?

## Ampliación del aprendizaje

Usa estas actividades para ampliar el aprendizaje de los estudiantes. Se pueden usar como actividades extras fuera del aula.

### Más Minecraft

Si ves que tu clase disfruta mucho del entorno *Minecraft*, [\*\*aquí tienes algunos enlaces a otros juegos de Minecraft que pueden jugar en línea\*\*](#). Estos juegos también enseñarán habilidades básicas de programación.



Esta obra está disponible bajo una [\*\*Licencia Creative Commons \(CC BY-NC-SA 4.0\)\*\*](#).

**Contáctanos** si desea contar con la licencia de los materiales de Code.org para uso comercial.

# Lección 14: Condicionales (Si/sino) con Abeja

55 minutos

## Resumen

Hasta ahora los/as estudiantes han estado escribiendo código que se ejecuta exactamente del mismo modo cada vez que se ejecuta - confiable, pero no muy flexible. En esta lección, su clase comenzará a codificar con condicionales, permitiéndoles escribir código que funcione de forma diferente dependiendo de las condiciones específicas que el programa encuentre.

## Propósito

Luego de la introducción a las condicionales vista en “Condicionales con cartas”, los estudiantes practicarán cómo aplicarlas en programas. Los bloques “si / si no” permitirán programas más flexibles. La abeja sólo recolectará néctar *si* hay una flor y hará miel sólo *si* hay un panal.

## Estándares

Curso Completo Alineamiento

### Estándares de Ciencias de la Computación CSTA K-12 (2017)

- ▶ **AP** - Algorithms & Programming

## Agenda

### Actividad previa (10 minutos)

Introducción

Revisión de desafíos en línea

### Actividad Principal (30 minutos)

Condicionales (Si/sino) con Abeja

### Actividad de cierre (15 minutos)

Reflexión

Ampliación del aprendizaje

## Objetivos

Los estudiantes podrán:

- Resolver desafíos usando una combinación de secuencias en bucle y condicionales.
- Traspasar declaraciones condicionales desde el lenguaje hablado hasta un programa.

## Preparación

- Juega con los rompecabezas para encontrar cualquier área potencialmente problemática para tu clase.
- Asegúrate de que cada estudiante tenga un diario de reflexión.

## Enlaces

**¡Aviso!** Por favor, haga una copia de cualquier documento que planea compartir con los estudiantes.

Para los estudiantes

- **Bloques sin conexión (cursos**

**C-F)** - Manipulativos

▼ Hacer una copia

## Vocabulario

- **Condicionales** - sentencias que solo se ejecutan bajo ciertas condiciones.

## Guía Didáctica

### Actividad previa (10 minutos)

#### Introducción

Repasa la actividad "Condicionales con cartas" con tus estudiantes.

- *¿Qué es una declaración condicional?*
- *¿Cuándo es útil una condicional?*
- *¿Cuáles son algunas de las condiciones que usaste en la última actividad?*

Ahora usaremos condicionales para ayudar a la abeja Code.org a lidiar con algunas nubes misteriosas. No sabemos si las flores tienen néctar o no, por lo que necesitaremos usar condicionales para asegurarnos de recolectar el néctar si hubiese y no intentar recolectarlo en flores donde no haya nada.

#### Revisión de desafíos en línea

Muestra un desafío de la lección. Recomendamos el desafío número 9.

- Pregunta que debería hacer la abeja cuando llegue a la nube.
  - La abeja debería usar una condicional para revisar si hay una flor o un panal.
- Use el bloque "si en flor / si no". Pregunta qué debería hacer la abeja si hay una flor. Si no hay una flor, habrá un panal, ¿qué debería hacer la abeja entonces?
  - Si hay una flor, la abeja debe "obtener néctar"; si hay un panal, la abeja debe "hacer miel".

Completa el resto del código y presiona "Ejecutar". Conversa con la clase por qué funciona el programa.

### Actividad Principal (30 minutos)

#### Condicionales (Si/sino) con Abeja

Estos desafíos podrían generar dudas y preguntas, por lo que es recomendable programar en parejas o implementar la regla de "pregúntale a 3 antes que a mí" (que los estudiantes le pidan ayuda al menos a tres compañeros antes de preguntarle al docente). Esto suscitará análisis y conversaciones que desarrollarán la comprensión de cada estudiante.



Vídeo: Condicionales: Sentencias "If"

-  2 Predicción
-  3-7 Desarrollo de Habilidades
  - 
  - 
  - 
  - 
  - 
-  8 Vídeo: Condicionales: Sentencias "If" e "If/Else"
-  9 Desarrollo de Habilidades
-  10 Predicción
-  11 Desafío
-  12-13 Práctica
  - 
  - 
-  14-15 Lecciones adicionales
  - 
  - 

## Actividad de cierre (15 minutos)

### Reflexión

#### Sugerencias:

- ¿Sobre qué se trataba la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- ¿Qué condicionales usaste hoy en tu código?
- ¿Qué otras condicionales podría usar una abeja? Algunos ejemplos son:
  - si hay árbol frente a mí, "zumbar" lejos de ahí.
  - si me duele un ala, descansar en tierra.
  - si veo otra abeja, decir "¡hola!".

## Ampliación del aprendizaje

Usa estas actividades para ampliar el aprendizaje de los estudiantes. Se pueden usar como actividades extras fuera del aula.

### Verdadero o falso

- Forma a los estudiantes como si fuesen a jugar el **juego del semáforo**.
- Escoge a un estudiante para que pase al frente como el locutor.
- El locutor elige una condición y les pide a todos que cumplan esa condición para poder dar un paso al frente.
  - Si tienes un cinturón rojo, da un paso al frente. - Si estás usando zapatillas, da un paso al frente.
- Intenten cambiar un poco las cosas, diciendo cosas como “si no eres rubio, da un paso al frente”.

### Anidando

- Divide a los estudiantes en grupos o en parejas.
- Pídeles que escriban declaraciones “si” para cartas en tiritas de papel, tales como: - El diseño es diamantes. - El color es rojo.
- Pídeles que hagan tiritas similares para los resultados. - Gana un punto. - Pierde un punto.
- Una vez que todo esté listo, pídeles que elijan tres tiritas de cada tipo y tres cartas, prestando atención al orden seleccionado.
- En tres hojas de papel, que los estudiantes escriban tres programas diferentes usando sólo las condicionales de las tiritas de papel que eligieron, en cualquier orden.
  - Anima a los estudiantes a poner declaraciones si dentro de otras declaraciones si.
- Ahora los estudiantes deben ejecutar los tres programas utilizando las cartas que sacaron, en el mismo orden para cada programa. - ¿Algunos programas dieron la misma respuesta? - ¿Alguno arrojó algo diferente?



Esta obra está disponible bajo una **Licencia Creative Commons (CC BY-NC-SA 4.0)**.

**Contáctanos** si desea contar con la licencia de los materiales de Code.org para uso comercial.

# Lección 15: Bucles 'mientras' en Granjera

55 minutos

## Resumen

En esta lección de **construcción de estudios**, los estudiantes estarán trabajando para llenar agujeros y cavar tierra en Farmer, pero no sabrán el tamaño de los agujeros ni la altura de los montículos de tierra. Para resolver estos acertijos, los estudiantes usarán un nuevo tipo de bucle.

## Propósito

Mientras los estudiantes continúan profundizando en la comprensión de los bucles, se toparán con problemas en que un comando necesitará ser repetido, pero se desconoce la cantidad de veces que necesitará serlo. Para eso existen los bucles mientras. En la lección de hoy, los estudiantes desarrollarán una comprensión básica de los bucles basados en condiciones, además de expandir sus conocimientos de los bucles en general.

## Estándares

Curso Completo Alineamiento

### Estándares de Ciencias de la Computación CSTA K-12 (2017)

- ▶ **AP** - Algorithms & Programming

## Agenda

### Actividad previa (10 minutos)

#### Introducción

#### Revisión de desafíos en línea

### Actividad Principal (30 minutos)

#### Bucles 'mientras' en Granjera

### Actividad de cierre (15 minutos)

#### Reflexión

## Objetivos

Los estudiantes podrán:

- Distinguir entre bucles que se repiten una cantidad fija de veces y bucles que se repiten mientras se cumpla una condición.
- Usar un bucle mientras para desarrollar programas que resuelvan problemas con valores desconocidos.

## Preparación

- Juega con los rompecabezas para encontrar cualquier área potencialmente problemática para tu clase.
- Asegúrate de que cada estudiante tenga un diario de reflexión.

## Enlaces

**¡Aviso!** Por favor, haga una copia de cualquier documento que planea compartir con los estudiantes.

Para los profesores

- **Programa ejemplo - Condicionales con cartas** - Guía de preparación para el docente

Para los estudiantes

- **Bloques sin conexión (cursos C-F)** - Manipulativos

## Vocabulario

- **Condición** - algo que un programa verifica para ver si es verdadero antes de permitir una acción.
- **Bucle** - la acción de hacer algo una y otra vez.
- **Repetir** - Hacer algo otra vez.
- **Bucle mientras** - Un bucle que se repite mientras se cumpla una condición.

## Guía Didáctica

### Actividad previa (10 minutos)

#### Introducción

Usa el término “mientras” en una oración en frente de su clase y pregúntales qué significa. Si dijese *“mientras haya un agujero, rellénenlo con tierra”*, ¿qué deberían hacer ellos? ¿Por cuánto tiempo lo harían?

Cuando se usa una palabra como “mientras”, se está estableciendo que la cantidad de tiempo que un bucle deba ejecutarse depende de una condición. Una condición es una declaración que es revisada para verificar si es verdadera o falsa. En el ejemplo que se dio, la condición es “si hay un agujero”. Sólo hay dos opciones, que haya o que no haya un agujero, por lo tanto la declaración sólo puede ser verdadera o falsa.

Comenta que van a aprender sobre un nuevo tipo de bucle. Anteriormente, los estudiantes usaron bucles sólo para repetir un comando una cantidad determinada de veces. En esta ocasión, no siempre sabrán cuántas veces se debe repetir el comando; sin embargo, si sabrán cuándo deben detenerse o cuándo deben seguir. Los bucles mientras le permiten al programador repetir un comando siempre que una condición se cumpla. Así, en el ejemplo anteriormente dado, la condición sería “la existencia de un agujero”.

Si hay tiempo, pide a los estudiantes conversar y analizar sobre otras circunstancias en que sería útil usar un bucle mientras. Algunos ejemplos son:

- Correr hacia una pelota mientras esté frente a ti.
- Llenar un vaso mientras haya espacio para más líquido.
- Caminar hacia adelante mientras haya un camino.

### Revisión de desafíos en línea

Elige uno de los desafíos de la lección. Recomendamos el desafío número 6.

- Pregúntale a la clase qué debe hacer la granjera cuando llegue a la pila de tierra.
  - Debe usar un bucle mientras para empezar a remover la tierra.
- Usa el bloque `mientras haya una pila / haz`. Pregunta qué debe hacer la granjera dentro del bucle mientras.

- La granjera debe `eliminar 1`. La granjera seguirá “eliminando 1 tierra” mientras haya tierra. En otras palabras, `elimina 1` dejará de ejecutarse cuando ya no quede tierra!

Completa el resto del código y presiona `Ejecutar`. Conversa con la clase por qué funciona el programa.

## Actividad Principal (30 minutos)

### Bucles 'mientras' en Granjera

Los bucles mientras normalmente no suponen un problema para los estudiantes; sin embargo, si consideras que podrían ser complicados para tu clase, recomendamos la programación en parejas. Esto permitirá un intercambio de ideas entre los estudiantes antes de implementar un código. La programación en parejas produce un aumento en la confianza y la comprensión de temas como los bucles mientras.

 1-3

Desarrollo de Habilidades

1

2

3

 4

Vídeo: Bucles "While" con la Granjera

 5

Predicción

 6-9

Desarrollo de Habilidades

6

7

8

9

 10

Desafío

 11-12

Práctica

11

12

 13

Predicción

 14

Desafío

## Actividad de cierre (15 minutos)

### Reflexión

El acto de escribir en sus diarios sobre lo aprendido, respecto de si les pareció útil y de lo que sintieron, ayuda a tus estudiantes a fortalecer cualquier conocimiento que hayan obtenido hoy y servir como un resumen al que puedan recurrir en el futuro.

#### Sugerencias:

- ¿Sobre qué se trataba la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- ¿Cuál es la diferencia entre un bucle mientras y un bucle repetir normal?
- Da un ejemplo de un desafío en dónde usarías un bucle mientras, pero no un bucle repetir. ¿Puedes dar un ejemplo de un desafío en dónde usarías un bucle repetir, pero no un bucle mientras?



Esta obra está disponible bajo una [Licencia Creative Commons \(CC BY-NC-SA 4.0\)](https://creativecommons.org/licenses/by-nc-sa/4.0/).

**Contáctanos** si desea contar con la licencia de los materiales de Code.org para uso comercial.

# Lección 16: Bucles 'hasta' en el Laberinto

55 minutos

## Resumen

En esta lección, los estudiantes aprenderán acerca de los bucles `hasta`. Crearán programas en los cuales el personaje repetirá acciones `hasta` que llegue al punto de detención deseado.

## Propósito

Esta serie de desafíos servirá para reforzar y desarrollar el conocimiento de bucles, añadiendo la condicional `hasta`. Al juntar estos dos conceptos, los estudiantes serán capaces de explorar el potencial de crear programas complejos e innovadores.

## Estándares

Curso Completo Alineamiento

### Estándares de Ciencias de la Computación CSTA K-12 (2017)

- ▶ **AP** - Algorithms & Programming

## Agenda

### Actividad previa (10 minutos)

#### Introducción

### Actividad Principal (30 minutos)

#### Bucles 'hasta' en el Laberinto

### Actividad de cierre (15 minutos)

#### Reflexión

## Objetivos

Los estudiantes podrán:

- Desarrollar programas comprendiendo múltiples estrategias para implementar condicionales.
- Traspasar declaraciones condicionales y bucles desde el lenguaje hablado hasta un programa.

## Preparación

- Juega con los rompecabezas para encontrar cualquier área potencialmente problemática para tu clase.
- Asegúrate de que cada estudiante tenga un diario de reflexión.

## Vocabulario

- **Condición** - algo que un programa verifica para ver si es verdadero antes de permitir una acción.
- **Condicionales** - sentencias que solo se ejecutan bajo ciertas condiciones.
- **Bucle** - la acción de hacer algo una y otra vez.
- **Repetir** - Hacer algo otra vez.
- **Hasta** - Un comando que establece realizar una acción sólo hasta que algo sea verdadero.

# Guía Didáctica

## Actividad previa (10 minutos)

### Introducción

En esta lección, los estudiantes desarrollarán bucles que se ejecutarán "hasta" que una condición sea verdadera. Ayúdalos a entender este concepto guiándolos con actividades grupales y proponiéndoles hacer algo hasta que una condición sea verdadera. Por ejemplo, que los estudiantes toquen su nariz "hasta" que les digas que paren.

### Revisión de desafíos en línea

Elige un desafío de la lección. Recomendamos el desafío número 4.

- Pregúntale a la clase qué debería repetir el ave para llegar al cerdo.
  - El ave debería repetir "avanzar", "girar a la derecha", "avanzar" y luego "girar a la izquierda".
- Pregunta qué podrían usar para repetir este código.
  - El ave debe repetir este patrón *hasta* llegar al cerdo.

Completa el resto del código usando el bucle "repetir hasta" y presiona "Ejecutar". Conversa con la clase por qué funciona el programa.

## Actividad Principal (30 minutos)

### Bucles 'hasta' en el Laberinto

#### Desafíos en línea

Relacionar los conceptos no es fácil; sin embargo, estas lecciones buscan ayudar a los estudiantes a ver las posibilidades infinitas para codificar cuando se usan condicionales. Si los estudiantes tienen dificultades para entender las similitudes o diferencias entre los bucles "hasta" y los bucles "mientras", pídeles que intenten pensar cómo usarían declaraciones similares en la vida real.



1

Desarrollo de Habilidades



2

Vídeo: Instrucciones 'Repetir Hasta'



3

Predicción



4-8

Desarrollo de Habilidades

4

5

6

7

8



Desafío



Práctica



Predicción

## Actividad de cierre (15 minutos)

### Reflexión

#### Sugerencias:

- ¿Sobre qué se trataba la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- ¿Cuál es la diferencia entre un bucle "hasta" y un bucle "mientras"?



Esta obra está disponible bajo una [Licencia Creative Commons \(CC BY-NC-SA 4.0\)](https://creativecommons.org/licenses/by-nc-sa/4.0/).

**Contáctanos** si desea contar con la licencia de los materiales de Code.org para uso comercial.

# Lección 17: Proyecto de fin de curso

45 minutos

## Resumen

¡Esta lección de **proyecto** lleva a los estudiantes a través del proceso de diseñar, desarrollar y exhibir nuevos proyectos!

## Propósito

Esta lección les brinda a los estudiantes la oportunidad de desarrollar un proyecto diseñado por ellos mismos, a través de un proceso paso a paso que requiere planificación y fomenta la creatividad.

## Estándares

Curso Completo Alineamiento

### Estándares de Ciencias de la Computación CSTA K-12 (2017)

- ▶ **AP** - Algorithms & Programming
- ▶ **IC** - Impacts of Computing

## Agenda

### Actividad previa (10 minutos)

#### Planificación

### Actividad Principal (25 minutos)

#### Proyecto de fin de curso

### Actividad de cierre (10 minutos)

#### Exhibición

## Objetivos

Los estudiantes podrán:

- Superar obstáculos como limitaciones de tiempo o errores.

## Preparación

- Realiza los desafíos para encontrar cualquier área potencialmente problemática para tu clase.
- Imprime una **\*Guía de planificación del proyecto** para cada estudiante (o para cada pareja).
- (Opcional) Completa tu propia guía de planificación y codifica tu propio proyecto para mostrárselo a los estudiantes.

## Enlaces

**¡Aviso!** Por favor, haga una copia de cualquier documento que planea compartir con los estudiantes.

Para los estudiantes

- **Guía de planificación - Proyecto del curso D** - Guía de Proyecto

▼ Hacer una copia

# Guía Didáctica

## Actividad previa (10 minutos)

### Planificación

¡Motiva y prepara a los estudiantes para la actividad de hoy!

#### Observaciones

Ya tuvimos la oportunidad de crear varios proyectos diferentes. La actividad que experimentarán hoy será mucho más abierta, ¡por lo que requerirá planificación de antemano! La planificación es una parte fundamental en la programación de un juego o de cualquier software. Por ende, antes de pasar a los computadores, destinaremos algo de tiempo a planificar los proyectos que queremos desarrollar.

**Distribuir:** entrega una **\*Guía de planificación de proyectos** a cada estudiante o pareja. Junto con los estudiantes, revisa los puntos de la guía y luego permíteles completarlos. Si es necesario, recurre al ejemplo.

#### Consejo didáctico

Si en esta actividad los estudiantes programarán en parejas, esta actividad inicial es una gran oportunidad para que practiquen cómo compartir y respetar las ideas de otros. Asegúrate de que los estudiantes estén siguiendo las normas para trabajar en equipo que hayas puesto para la clase. De lo contrario, explícales brevemente lo que esperas de ellos.

## Actividad Principal (25 minutos)

### Proyecto de fin de curso

Con sus guías de planificación a mano, los estudiantes están listos para que sus proyectos vean la luz. Estos niveles se corresponden con la estructura de la guía de planificación y ayudan a los estudiantes a abordar el proceso de transformar sus ideas en código.

#### Consejo didáctico

Los estudiantes experimentarán muchos desafíos y cometerán errores mientras programan. Es muy probable que sus proyectos se vuelvan versiones truncadas de su visión original. Recuérdales que esta clase de compromiso es normal en el diseño de software. ¡Está bien si no pueden desarrollar todas las características que planearon!



1

Ejemplo de proyectos



2

Crear tu proyecto

## Actividad de cierre (10 minutos)

### Exhibición

Para celebrar el trabajo de tus estudiantes, en los últimos 10 minutos (o más) permíteles exponer sus proyectos. Esto puede hacerse de varias maneras. Estas son algunas sugerencias:

- **Demostración al público:** elige a algunos voluntarios ejemplares para enseñar brevemente sus proyectos frente a la clase. Mientras lo hacen, pídeles compartir cómo fue la experiencia de planificar y codificar para ellos y que comenten ideas que les hubiese gustado implementar.
- **Prueba en parejas:** pide a los estudiantes o grupos que se dividan en parejas y prueben los proyectos de los demás. Mientras lo hacen, pídeles que proporcionen una crítica positiva y constructiva a su compañero. El beneficio de esto es que tendrán la oportunidad de hacer, recibir y responder a las críticas en un contexto más ameno y amigable.
- **Exhibición:** asegúrate de que todos los estudiantes tengan sus proyectos listos para ser probados. Usando un método de “sillitas musicales”, pídeles que se muevan de un computador a otro probando los proyectos por algunos minutos, hasta que des una señal para que cambien de estación. Repite esto por algunos minutos. A pesar de haber menos oportunidades para una comunicación estructurada, este método asegura que tus estudiantes probarán tantos proyectos como sea posible.



Esta obra está disponible bajo una [Licencia Creative Commons \(CC BY-NC-SA 4.0\)](https://creativecommons.org/licenses/by-nc-sa/4.0/).

**Contáctanos** si desea contar con la licencia de los materiales de Code.org para uso comercial.