

# Curso Express

Aprende los fundamentos de las ciencias de la computación y la seguridad en Internet. Al final del curso, crea tu propio juego o historia para compartir.

Enlaces para docentes: [Teacher Videos Playlist][0]

## Lección 1: Programación: Programación en Papel Cuadriculado

Unplugged | Programming | Program

¡En esta lección, programarás a tu amigo/a para dibujar!

## Lección 2: Introducción a los Desafíos Online

Sequencing | Debugging | Loop | Angry Bird | Maze | Artist

Las lecciones de la 1-9 son consideradas más elevadas. Si siente que las primeras actividades fueron muy simples para su clase, sienta la libertad de escoger sus actividades favoritas o sáltese hasta la número 10.

## Lección 3: Construyendo los Cimientos

Unplugged | Persistence | Frustration

## Lección 4: Debugging en Scrat (Actualizado)

Debugging | Ice Age

## Lección 5: Bucles: mis amigos robóticos

Unplugged | Loop | Repeat

¡En esta actividad usted "programa" sus amigos usando loops para apilar vasos para hacer torres geniales!

## Lección 6: Bucles en Artista

Loop | Artist

¡Estos desafíos online te permitirán dibujar algunos diseños asombrosos usando loops!

## Lección 7: Loops Anidados

Nested Loops | Loops

## Lección 8: Bucles anidados en Frozen

Loop | Nested Loop | Artist

Aquí estarás usando loops anidados para hacer proyectos geniales para mostrar a tu familia y amigos.

## Lección 9: Más allá de la Programación: Internet

Unplugged | Internet

¿Te has preguntado alguna vez cómo viaja la información por Internet? Esta lección te enseñará los elementos básicos de cómo funciona Internet.

## Lección 10: Ciudadanía Digital: Practicando Ciudadanía Digital

Common Sense Education | Unplugged

Algunas informaciones no son seguras para compartir online. Esta lección te ayudará a aprender la diferencia entre información segura y privada.

## Lección 11: Ciudadanía Digital: Sacar al Malo/a de la Pantalla

Common Sense Education | Cyberbullying | Unplugged

Esta lección te enseñará sobre cosas malas que ocurren online y cómo responder a ellas.

## Lección 12: Eventos: El gran evento

Unplugged | Events

¡Esta lección te enseñará sobre eventos, que son necesarios para los videojuegos que tú juegas!

## Lección 13: Eventos en Star Wars

Star Wars | Event

Esta lección te guiará haciendo tu propio videojuego de Star Wars.

## Lección 14: Eventos con Flappy

Event | Flappy

¡Aquí harás un súper videojuego con Flappy!

## Lección 15: Eventos en Bounce

Event | Bounce

¿Alguna vez has pensado poder jugar videojuegos en la escuela? ¡En esta lección, harás tu propio videojuego!

## Lección 16: Condicionales: condicionales con naipes

Conditionals | Unplugged

¡Es hora de jugar un juego donde ganas puntos sólo bajo ciertas condiciones!

## Lección 17: Loops Mientras en la Granjera

While Loops | Loops | Farmer

Los loops son tan útiles en la codificación. Esta lección te enseñará sobre un nuevo tipo de loops: ¡loops mientras!

## Lección 18: Condicionales & Loops en Laberinto

Conditional | Loop | Maze | Angry Bird | Zombie

¡Puedes hacer algunas cosas asombrosas cuando usas condicionales y loops juntos!

## Lección 19: Condicionales en Minecraft

Conditional | Minecraft

¡Esquiva la lava! Aquí aprenderás acerca de los condicionales en el mundo de Minecraft.

## Lección 20: Condicionales & Loops en Cosechadora

Conditional | Loop | Harvester

No siempre está claro cuándo usar cada condicional. Esta lección te ayudará a tener práctica en decidir qué hacer.

## Lección 21: Variables: Variables en Sobres

Unplugged | Variable

Los sobres y las variables tienen algo en común: ambos pueden contener cosas valiosas. Aquí aprenderás lo que son las variables y las cosas geniales que éstas pueden hacer.

## Lección 22: Variables en Artista

Variable | Artist

¡No te olvides de traer la creatividad a la clase! En estos desafíos harás dibujos fantásticos usando variables.

## Lección 23: Variables en Play Lab

Variable | Play Lab

¡Pronto aprenderás a hacer que los personajes interactúen en un juego usando variables!

## Lección 24: Loops Para: Diversión con Loops Para

Unplugged | For Loops

¡Vas a entretenerte muchísimo aprendiendo sobre los Loops "Para"!

## Lección 25: Loops Para en Abeja

For Loop | Bee

Bsss...bssss... En estos desafíos guiarás a una abeja hacia el néctar y la miel usando Loops "Para".

## Lección 26: Loops Para en Artista

For Loop | Artist

Prepárate para hacer tu próxima obra maestra. Aquí usarás los loops "para" para hacer dibujos que te dejarán con la boca abierta.

## Lección 27: Funciones: composición de canciones con parámetros

Unplugged | Function | Parameter

¡Podrías lanzar el próximo gran éxito! En esta lección, aprenderás qué son los parámetros y cómo hacen fantásticas canciones.

## Lección 28: Funciones en Abeja

Function | Bee

¡La abeja necesita tu ayuda otra vez! Aquí usarás funciones para conseguir néctar y hacer miel.

## Lección 29: Funciones con Parámetros en Artista

Function | Parameter | Artist

Prepara tus dedos programadores. En estos desafíos harás dibujos impresionantes en Artista usando funciones con parámetros.

## Lección 30: Funciones con Parámetros en Abeja

Function | Parameter | Bee

Ya has podido practicar un poco usando funciones con parámetros. ¡Esta lección continuará tu práctica con Abeja!

## Lección 31: Explorar Ideas de Proyectos

Project | Define | Prepare | Try | Revise | Reflect

¡Es hora de inspirarse! Estos desafíos te mostrarán muchos juegos e ilustraciones pre armadas que te ayudarán a desarrollar tu plan para el GRAN proyecto.

## Lección 32: El Proceso de Diseño

Project

Proyectos así de grandes toman tiempo mucha planificación. Aquí, aprenderás sobre el proceso de diseño que usarás para construir tu propia creación.

## Lección 33: Construye Tu Proyecto

Project

¡Finalmente, puedes empezar a construir tu proyecto!

## Lección 34: Revisa Tu Proyecto

Project

Roma no se construyó en un día y tu proyecto tampoco debería. Tómate el tiempo para editar y revisar tu proyecto para que sea lo mejor que pueda ser.

## Lección 35: Presenta Tu Proyecto

Project

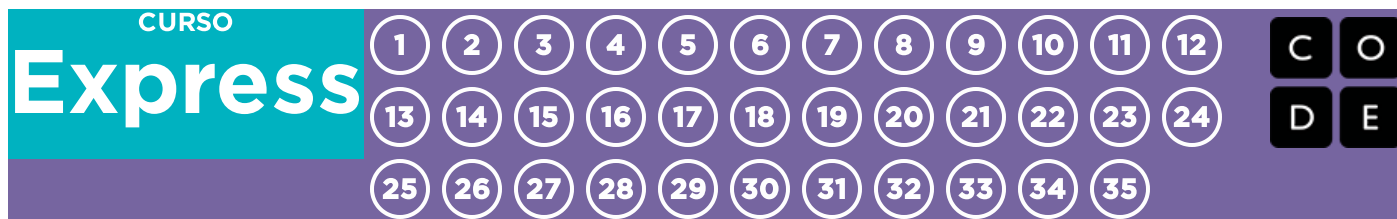
¡Es hora de mostrar tu trabajo! Aquí presentarás tu fantástico proyecto a tus pares.



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



# Lección 1: Programación: Programación en Papel Cuadriculado

Sin conexión | Programación | Programa

## Descripción

Al “programarse” entre sí para dibujar imágenes, los estudiantes comenzarán a comprender de qué se trata la programación. Para empezar, los estudiantes se darán instrucciones mutuamente para colorear cuadrados en un papel cuadriculado, con la intención de reproducir un dibujo ya existente. Si hay tiempo, la lección puede concluir con imágenes que los estudiantes creen por sí mismos.

## Propósito

El objetivo de esta actividad es desarrollar las habilidades de pensamiento crítico y el entusiasmo por el curso.

Al presentar conceptos básicos como *programación* y *algoritmos* a la clase en una actividad sin conexión, los estudiantes que se sienten intimidados por las computadoras pueden comenzar a comprender estos temas. La programación y los algoritmos son esenciales en las ciencias de la computación. En esta lección, los estudiantes aprenderán a transformar las instrucciones en un programa y reconocer un algoritmo.

## Agenda

### Actividad previa (20 min)

Vocabulario

Introducción a la Programación con Papeles Cuadriculados.

### Actividad Principal (20 min)

Programación con Papel Cuadriculado - Ficha

### Actividad de Cierre (15 min)

Conversación Rápida: ¿Qué aprendimos?

Escribir en el Diario

### Evaluación (10 min)

Programación con Papel Cuadriculado - Evaluación

### Aprendizaje Extendido

## Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Entender la dificultad de convertir problemas reales en programas.
- Organizar los eventos secuenciales en su orden lógico. Explicar cómo las ideas pueden ser claras y, aún así, ser malinterpretadas por una computadora.
- Practicar la comunicación de ideas a través de códigos y símbolos.

## Preparación

- ☐ Vea **Programación con Papel Cuadriculado - Video para el docente.**
- ☐ Vea **Programación con Papel Cuadriculado - Video de lección en acción.**
- ☐ Imprima una copia **Programación con Papel Cuadriculado - Ficha** para cada grupo.
- ☐ Imprima una copia **Programación con Papel Cuadriculado - Evaluación** para cada estudiante.
- ☐ Entregue a cada grupo copias suficientes de papeles cuadriculados, lápices y lapiceras.
- ☐ Asegúrese que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones.**

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Programación con Papel Cuadriculado - Video para el docente**
- **Programación con Papel Cuadriculado - Video de lección en acción**

- **Programación con Papel Cuadriculado** -  
Guía de respuestas en fichas  
[Hacer una copia ▼](#)
- **Programación con Papel Cuadriculado** -  
Guía de respuestas de la evaluación  
[Hacer una copia ▼](#)

Para los estudiantes

- **Programación con Papel Cuadriculado** -  
Video sin conexión (**descargar**)
- **Programación con Papel Cuadriculado** -  
Ficha
- **Programación con Papel Cuadriculado** -  
Evaluación [Hacer una copia ▼](#)

## Vocabulario

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

# Guía docente

## Actividad previa (20 min)

### Vocabulario

Esta lección tiene dos palabras nuevas:

- **Algoritmo** - Díganlo conmigo: Al-go-rit-mo.

Una lista de pasos que puedes seguir para finalizar una tarea.

- **Programa** - Díganlo conmigo: Pro-gra-ma.

Un algoritmo que ha sido codificado en algo que puede ser leído por una máquina.

### Introducción a la Programación con Papeles Cuadriculados.

En esta actividad, nos vamos a guiar mutuamente para hacer dibujos, sin dejar que los/as otros/as vean el dibujo original.

Para este ejercicio, usaremos hojas cuadriculadas de papel de 4x4. Comenzando en la esquina superior izquierda, guiaremos la Máquina de Realización Automática (ARM por sus siglas en inglés) de nuestros/as compañeros/as de equipo con instrucciones simples. Estas instrucciones incluyen:

- Moverse un Cuadrado a la Derecha.
- Moverse un Cuadrado a la Izquierda.
- Moverse un Cuadrado hacia Arriba.
- Moverse un Cuadrado hacia Abajo.
- Llenar el Cuadrado con Color.

Por ejemplo, aquí está cómo escribiríamos un algoritmo para dar la instrucción a un amigo/a (quien simula ser una máquina que dibuja) para colorear su hoja cuadriculada, para que luzca como la imagen siguiente.

#### 💡 Lesson Tip

Pida a la clase que se imagine que su brazo es una Máquina de Realización Automática (ARM por sus siglas en inglés). La idea de los “algoritmos” y los “programas” cobrará más vida incluso si sus estudiantes sienten que están realmente controlando sus movimientos.

## Actividad Principal (20 min)

### Programación con Papel Cuadriculado - Ficha

- Divida a los/as estudiantes en parejas.
- Haga que cada pareja elija una imagen de la hoja de trabajo.
- Discuta el algoritmo para dibujar esa imagen con el/la compañero/a.
- Conviertan el algoritmo en un programa usando los símbolos.
- Intercambien los programas con otra pareja y dibujen la imagen de la otra pareja.
- ¡Escojan otra imagen y repítanlo

## Actividad de Cierre (15 mín)

### Conversación Rápida: ¿Qué aprendimos?

- ¿Qué aprendimos hoy?
- ¿Qué pasaría si usamos las mismas flechas, pero las reemplazáramos con "Cuadros para llenar" con "Poner Ladrillo"? ¿Qué podría hacer?
- ¿Qué más podríamos programar si sólo cambiáramos lo que significan las flechas?



## Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil, y cómo se sienten. Puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- Dibuja otra imagen que podrías codificar. ¿Puedes escribir el programa para hacer este dibujo?
- ¿Cuáles son otras instrucciones que podrían ser útiles para imágenes grandes o complicadas?

## Evaluación (10 min)

### Programación con Papel Cuadriculado - Evaluación

- Distribuya la hoja de la evaluación y permita que los/as estudiantes completen la actividad en forma independiente luego que las instrucciones queden claras.
- Esto debería serles familiar debido a las actividades en lecciones previas.

## Aprendizaje Extendido

Use estas actividades para mejorar el aprendizaje de sus estudiantes. Pueden usarse como actividades fuera del aula u otros enriquecimientos.

### Mejor y Mejor

- Pida a la clase que intente inventar sus propias imágenes.
- ¿Pueden descifrar cómo programar las imágenes que crearon?

### Desafío para la Clase

- Como profesor/a, dibuje una imagen en una cuadrícula de 5x5.
- ¿Puede la clase codificarla con usted?

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

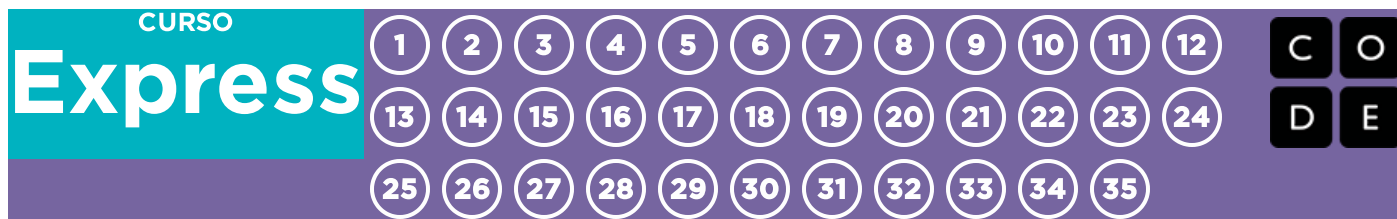
- **AP** - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



# Lección 2: Introducción a los Desafíos Online

Secuencia | Depurar | Bucle | Angry Bird | Laberinto | Artista

## Descripción

En esta progresión, los/as estudiantes comenzarán con una introducción (o revisión dependiendo en la experiencia de su clase) del espacio de trabajo online de Code.org. Aprenderán las funcionalidades básicas de la interfaz, incluyendo los botones Ejecutar, Reiniciar, y Paso. También se introduce al inicio del video Arrastrar, Eliminar, y los bloques conectores Blockly. En los desafíos, practicarán sus habilidades de secuenciación y debugging en Laberinto y Artista.

## Propósito

Reconocemos que cada aula tiene un espectro de comprensión de cada materia. Algunos/as estudiantes de su clase pueden ser magos/as de la computación, mientras otros/as no han tenido mucha experiencia. Con el propósito de crear un campo de juego (y aprendizaje) igualitario, hemos desarrollado esta “Etapa de Reforzamiento” para el Curso E. Puede ser usado como introducción o revisión de cómo usar Code.org y conceptos básicos de la ciencia de la computación. Esta etapa, en conjunto con las tres siguientes, cubren todos los pre-requisitos necesarios para empezar el Curso E.

## Agenda

### Actividad previa (15 mín)

Introducción  
Vocabulario

### Actividad Puente - Programación (15 mín)

Actividad desconectada usando bloques de papel  
Vista previa de desafíos en línea como una clase

### Actividad Principal (30 min)

Curso Rápido Fundamentos de Ciencias de la Computación  
- Sitio web

### Actividad de Cierre (15 mín)

Escribir en el Diario

### Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Ordenar comandos de movimiento como pasos secuenciales en un programa.
- Modificar un programa existente para solucionar errores.
- Dividir una secuencia larga de instrucciones en la secuencia repetible más grande.

## Preparación

- ☐ Juegue los desafíos **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web** para encontrar áreas de potenciales problemas para su clase.
- ☐ Asegúrese que cada estudiante tenga un [think-spot-journal][2].

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso Rápido Fundamentos de Ciencias de la Computación** - Sitio web
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación** - Recomendaciones de lecciones [Hacer una copia ▾](#)

Para los estudiantes

- **Bloques Blockly Sin Conexión (Cursos 2-5)** - Manipuladores
- **Diario Think Spot** - Diario de reflexiones [Hacer una copia ▾](#)

## Vocabulario

- **Bug o error** - parte de un programa que no funciona correctamente.
- **Depurar** - 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 docente

## Actividad previa (15 mín)

### Introducción

Los/as estudiantes aprenderán muchos conceptos nuevos o revisarán muchos conceptos básicos. De acuerdo con la experiencia de su clase, puede cubrir el siguiente vocabulario o pasar a una actividad de puente. Recomendamos usar las siguientes palabras en oraciones si las definiciones no están cubiertas explícitamente.

### Vocabulario

Esta lección tiene cuatro palabras nuevas e importantes de vocabulario:

- **\*\* Programa \*\*** - Dilo conmigo: Pro - Gra - ma      Un algoritmo que se ha codificado en algo que puede ser ejecutado por una máquina.
- **\*\* Programación \*\*** - Dilo conmigo: Pro - Gra - ma - ción      El arte de crear un programa.
- **\*\* Bug \*\*** - Dilo conmigo: Bug      Un error en un programa que impide que el programa se ejecute como se espera.
- **\*\* Debugging \*\*** - Dilo conmigo: De - Bugging      Búsqueda y corrección de errores en programas.
- **\*\* Loop \*\*** - Dilo conmigo: Loop      La acción de hacer algo una y otra vez.

## Actividad Puente - Programación (15 mín)

Esta actividad ayudará a llevar los conceptos desconectados de "Programación en papel cuadriculado" al mundo en línea en el que los estudiantes se están moviendo. Elige *una* de las siguientes opciones para hacer con su clase:

### Actividad desconectada usando bloques de papel

Al igual que en "Programación en papel cuadriculado", haz que los estudiantes de tu clase se pongan en parejas. Pase varios bloques llenar 1 y mover\_\_\_ de **Bloques Blockly Sin Conexión (Cursos 2-5) - Manipuladores** a cada pareja. Haz que cada pareja de estudiantes dibuje un diseño en una gráfica de cuatro por cuatro de [csf-graph-paper-programming-worksheet]. Luego, haz que los estudiantes trabajen juntos para escribir el programa necesario para dibujar este diseño utilizando los bloques de papel de Blockly. Los estudiantes deberán escribir *arriba*, *abajo*, *derecha* o *izquierda* en el bloque *mover* \_\_\_\_\_. ¡Asegúrate de que los estudiantes sepan que el programa va de arriba a abajo y que los bloques deben tocarse!

### Vista previa de desafíos en línea como una clase

Toma un desafío de **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**. Recomendamos el desafío 6 para esta actividad. Divida a los estudiantes en grupos de tres o cuatro. Pídeles que "programen" Scrat, la ardilla de La Era del Hielo, para llegar a la bellota con las flechas de "Programación en papel cuadriculado".



La clase no necesitará usar la última flecha.

Una vez que todos los grupos tengan una respuesta, discute el camino con su clase.

## Actividad Principal (30 min)

# Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

Los docentes juegan un rol vital en la educación de la ciencia de la computación y en la contribución de un ambiente de clases colaborativo y dinámico. Durante las actividades en línea, el rol del docente es principalmente el de motivar y brindar apoyo. Las lecciones en línea están diseñadas para centrarse en los estudiantes, por lo que los docentes deben evitar meterse cuando sus estudiantes están estancados. Algunas ideas sobre cómo hacer esto son:

- Utilizar la programación en parejas cada vez que sea posible durante la actividad.
- Motivar a los estudiantes con preguntas/desafíos que deben comenzar por preguntar a sus compañeros.
- Las preguntas sin responder pueden ser escaladas a un grupo cercano, quienes pueden ya tener la respuesta.
- Recuerde a los estudiantes que pueden utilizar el proceso de depuración antes de que usted se acerque a ayudarles.
- Haga que sus estudiantes describan el problema que están viendo. ¿Qué se supone que haga? ¿Qué está haciendo? ¿Qué te dice eso?
- Recuerde a los estudiantes que se sientan frustrados que la frustración es un paso en el camino hacia el aprendizaje, y que la perseverancia valdrá la pena.
- Si un estudiante sigue estancado después de todo esto, haga preguntas que los lleven a descubrir el error por sus propios medios.

## 💡 Recomendación para el profesor

Muestra a los estudiantes la forma **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
- Asegúrate de que el compañero le pueda describir la solución en voz alta antes de que te retires.

## Actividad de Cierre (15 mín)

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil y cómo se sienten, puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el Diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- ¿Qué significa "programar"?
- ¿Por qué es importante la programación?
- ¿Qué es lo que le gustaría aprender más acerca de los computadores?

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

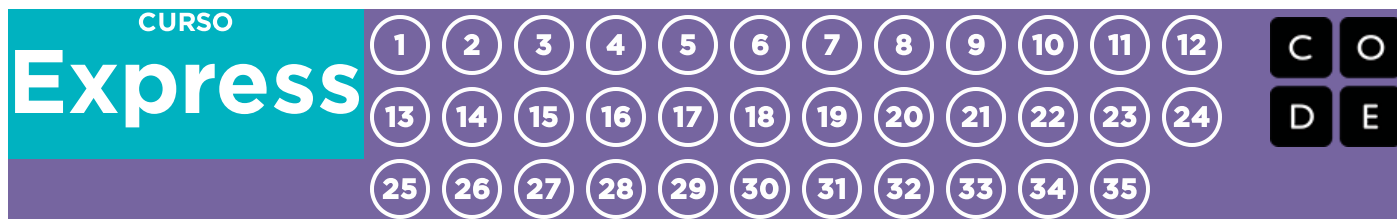
► AP - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



# Lección 3: Construyendo los Cimientos

Sin conexión | Perseverancia | Frustración

## Descripción

Los problemas nuevos y sin resolver, son a menudo muy difíciles. Si queremos tener alguna posibilidad de hacer algo creativo, útil e inteligente, entonces necesitamos estar dispuestos a encarar los problemas difíciles incluso si esto significa fracasar algunas veces antes de tener éxito. En esta lección, los/as estudiantes construirán una estructura con materiales comunes. La estructura será probada en su habilidad de sostener un libro por más de diez segundos. La mayoría de los/as estudiantes no podrán lograrlo la primera vez, pero es importante que sigan adelante y continúen intentándolo.

## Propósito

Esta lección enseña que el fracaso no es el fin del viaje, sino un indicio de cómo tener éxito. La mayoría de los/as estudiantes se sentirán frustrados/as en algún momento de esta lección, pero es importante enfatizar que el fracaso y la frustración son pasos comunes hacia la creatividad y el éxito.

## Agenda

### Actividad previa (20 min)

Vocabulario

Intentar, Intentar de Nuevo

### Actividad Principal (20 min)

Construyendo los Cimientos

### Actividad de Cierre (10 min)

Charla Exprés: ¿Qué aprendimos?

Escribir en el Diario

### Aprendizaje Extendido

## Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Describir los pasos para completar un desafío de ingeniería estructural.
- Predecir y analizar posibles problemas en la creación de la estructura.
- Crear una estructura basada en el plan del equipo.

## Preparación

☐Vea **Construyendo los Cimientos - Video para el docente**.

☐Vea el **Construyendo los Cimientos - Video de lección en acción**.

☐Imprima **Construyendo los Cimientos - Guía de preparación para el docente**.

☐Junte suficientes elementos de construcción (marshmallows o gomitas con mondadientes o palitos de helado) para cada grupo. No tiene que dar ninguna cantidad fija, solamente asegúrese de poner límite a la cantidad de materiales.

☐Entregue un **Diario Think Spot - Diario de reflexiones** a cada estudiante.

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Construyendo los Cimientos** - Video sin conexión ([descargar](#))
- **Construyendo los Cimientos** - Video para el docente
- **Construyendo los Cimientos** - Video de lección en acción
- **Construyendo los Cimientos** - Guía de preparación para el docente

Para los estudiantes

- **Diario Think Spot** - Diario de reflexiones

Hacer una copia ▼

## Vocabulario

- **Frustrado** - sentirse molesto o enojado porque algo no sale como uno quiere.
- **Perseverancia** - intentar algo una y otra vez, incluso si es muy difícil.

# Guía docente

## Actividad previa (20 min)

### Vocabulario

Esta lección tiene una palabra nueva: Perseverancia – Díganlo conmigo: Per – se – ve – ran- cia Intentar una y otra vez, incluso cuando algo es muy difícil.

### Intentar, Intentar de Nuevo

- ¿Todos/as hicieron todo bien la primera vez?
- Cuando era un bebé aprendiendo a caminar, ¿Me paré y corrí en mi primer intento?
- Sometimes, the best and most useful things to do are the hardest to learn.
  - Puede tomar un tiempo aprender cosas difíciles.
  - Si no haces algo bien al principio, ¿Significa eso que nunca lo harás?
  - ¿Puedes pensar en algo que fue difícil al principio, pero que ahora puedes hacer de manera bastante fácil?
    - Caminar
    - Hablar
    - Andar en bicicleta
- Cuando fracasas al hacer algo, tienes un indicio de lo que salió mal. Sólo tienes que buscar.
  - Si tu bicicleta se vuelca, la próxima vez debes trabajar en el equilibrio.
  - Si estás inflando un globo y se revienta, la próxima vez necesitas menos aire.
- Piensa en los errores como oportunidades de aprender o cómo hacer algo mejor, la siguiente vez.

#### 💡 Lesson Tip

Aquí hay grandes recursos para preparar su clase con el concepto de la perseverancia. Antes, dejarlos trabajar en este proyecto:

- **Mouse Wants a Cracker**
- **Fall 7 Times, Stand Up 8**
- **Never Ever Give Up**
- **If You Quit Too Soon**

## Actividad Principal (20 min)

### Construyendo los Cimientos

¿Alguna vez has comenzado una tarea, para luego descubrir que era mucho más difícil de lo que pensaste que sería? Las tareas difíciles pueden hacernos querer renunciar, pero si nos mantenemos apegados a nuestro objetivo y continuamos intentándolo, entonces ¡podríamos hacer algo mejor de lo que nunca hemos hecho!

En este desafío, trabajaremos en construir torres que sean lo suficientemente fuertes para sostener un libro por al menos 10 segundos, usando materiales de la vida diaria.

Reglas:

- Usa sólo los materiales proporcionados para construir la torre.
- Las torres pueden tener cualquier forma, pero tienen que ser al menos del alto de un vaso de papel.
- La torre debe soportar el peso de un libro por 10 segundos.

#### Instrucciones:

1. Divida a los/as estudiantes en grupos de tres o cuatro.
2. Explique las reglas del desafío, mencionadas más arriba.
3. Proporcione a cada grupo una cantidad limitada de materiales e infórmeles que no podrán obtener más.

#### 💡 Lesson Tip

La etapa de planificación puede ser difícil para estudiantes jóvenes. Puede ser útil ubicar algunas ideas de “ejemplo” al frente del aula. No avise que están ahí. Simplemente, anime a los/as estudiantes a dar una vuelta si se frustran. Intente animarlos/as a ubicar las sugerencias por sí mismos, si es posible.



4. Desafíe a la clase a adelantarse al problema y planificar su método de construir su primera torre.
  5. Anime a los/as estudiantes a comenzar la construcción, y luego pídale que le avisen cuando crean que han logrado cumplir con el desafío descrito en las reglas.
  6. Pruebe cada estructura. ¿Es más alta que un vaso? ¿Sostiene el libro?
  7. Si no es así, haga que los/as estudiantes entren en un ciclo de planificación, arreglo, prueba, y planificación nuevamente, hasta que cumplan el desafío.
  8. Felicite a los/as estudiantes cuando tengan éxito y ¡tome fotos de las torres exitosas!
- coursee/building-a-foundation/Main Activity -> content File: Activity. json!

## Actividad de Cierre (10 min)

### Charla Exprés: ¿Qué aprendimos?

- ¿Te sentiste orgulloso/a de lo que hiciste?
- ¿Crees que podrías hacer una torre del alto de una silla que pudiera sostener a una persona?
  - ¿Cuántas gomas de mascar crees que necesitarías?
- ¿Hubo momentos en que pensaste darte por vencido/a?
  - ¿Cómo superaste ese sentimiento?

#### 💡 Lesson Tip

Las preguntas de la Charla Exprés están orientadas para que se produzca pensamiento general y saber cómo la lección se relaciona al gran mundo y al aún más gran futuro de los/as estudiantes. Use el conocimiento de su aula para decidir si quiere discutir esto como clase, en grupos, o con un/a compañero/a de banco.

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil, y cómo se sienten puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro. Proporcionamos un Diario Think Spot como una plantilla básica para que los/as estudiantes la usen como su diario.

#### Sugerencias para el diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- Haz un dibujo de tu estructura.
- ¿Cuáles fueron algunos problemas que encontraste durante la construcción? ¿Cómo solucionaste esos problemas?

## Aprendizaje Extendido

Use estas actividades para mejorar el aprendizaje de los/as estudiantes. Pueden usarse como actividades fuera del aula u otro tipo de enriquecimientos.

### ¡Inténtalo de Nuevo!

Intenta hacer la misma actividad con materiales diferentes.

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

- AP - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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

# Lección 4: Debugging en Scrat (Actualizado)

Depurar | Era de Hielo

## Descripción

Debugging es un elemento esencial para aprender a programar. En esta lección, los/as estudiantes encontrarán desafíos que han sido resueltos incorrectamente. Necesitarán recorrer el código existente para identificar errores, incluyendo loops incorrectos, bloques faltantes, bloques adicionales y bloques que están fuera de servicio.

## Propósito

Los/as estudiantes de tu clase pueden sentirse frustrados/as con esta lección debido a la esencia del debugging. Debugging es un concepto muy importante para la programación de computadores. Los/as informáticos/as tienen que ser muy buenos/as para enfrentarse a los errores de sus propios programas. Debugging obliga a los/as estudiantes a reconocer los problemas y a superarlos al mismo tiempo que desarrollan el pensamiento crítico y las habilidades de resolución de problemas.

## Agenda

### Actividad previa (15 mín)

Introducción  
Vocabulario

### Actividad Principal (30 min)

Curso E Desafíos Online - Sitio web

### Actividad de Cierre (5 - 10 minutos)

Escribir en el Diario

### Aprendizaje Extendido

### Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Predecir dónde fallará un programa.
- Modificar un programa existente para solucionar errores.
- Reflexionar sobre el proceso de depuración de manera apropiada para cada edad.

## Preparación

- ▣ Recorra el **Curso E Desafíos Online - Sitio web** para encontrar cualquier área potencial de problema para su clase.
- ▣ (Opcional) Elija un par de desafíos para hacer de forma grupal con su clase.
- ▣ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones**.
- ▣ Asegúrese de que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones**.
- ▣ Revise **Receta Debugging - Guía para el estudiante** con su clase.

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso Rápido Fundamentos de Ciencias de la Computación** - Sitio web
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación** - Recomendaciones de lecciones [Hacer una copia ▾](#)

Para los estudiantes

- **Diario Think Spot** - Diario de reflexiones [Hacer una copia ▾](#)

- **Receta Debugging** - Guía para el estudiante [Hacer una copia ▼](#)

## Vocabulario

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

# Guía docente

## Actividad previa (15 mín)

### Introducción

Pregunte a los/as estudiantes sobre problemas que tienen que resolver en su diario vivir.

- ¿Cómo arreglas algo que no está funcionando?
- ¿Sigues una serie de pasos específicos?
- Los desafíos en esta unidad ya han sido resueltos para ti (¡Yupi!), pero no parecen estar funcionando bien (¡buuu!).
- A los problemas en estos programas los llamamos “Bugs” y será tu trabajo resolverlos.

### Vocabulario

Esta lección tiene tres nuevas e importantes palabras de vocabulario:

- **Bug** - Repitan conmigo: Bag. Algo que está mal. Un error.
- **Debugging** - Repitan conmigo: Di-ba-gin. Encontrar y reparar errores.
- **\_Perseverancia\_** - Repitan conmigo: Per-se-ve-ran-cia. No rendirse. La perseverancia funciona mejor cuando intentas las cosas de diferentes formas, repetidas veces.

Diga:

Debugging es un proceso. Primero, deben reconocer que hay un error en su programa. A continuación, trabajen en el programa paso a paso para encontrar el error. Prueben el primer paso, ¿funcionó? Luego el segundo, ¿qué tal ahora? Si se aseguran de que todo funciona línea por línea, entonces cuando lleguen al lugar donde su código no está haciendo lo que se supone que debe hacer, saben que han encontrado un error. ¡Una vez que hayan descubierto su error, pueden trabajar para arreglarlo (o hacer "debugging")!

Si cree que logrará emocionar a la clase, puede presentar al personaje de los desafíos de hoy, Scrat de La Era del Hielo. Si los/as estudiantes no están familiarizados/as con Scrat, **show some videos** de la peculiar ardilla metiéndose en problemas.

## Actividad Principal (30 min)

### Curso E Desafíos Online - Sitio web

Antes de permitir que los/as estudiantes trabajen en el computador, recuérdelos las ventajas de **Programación en Parejas - Video para el estudiante** y de pedir ayuda a sus compañeros/as. Siente a los/as estudiantes en parejas y recomíéndelos que al menos le pregunten a dos compañeros/as, antes de recurrir a usted.

Como se mencionó en el propósito de esta lección, asegúrese que los/as estudiantes estén conscientes de que enfrentarán desafíos frustrantes. Dígales que está bien sentirse frustrados/as, pero que es importante resolver el problema y pedir ayuda. A medida que los/as estudiantes resuelven los desafíos, pasee por la sala, para asegurarse que ningún/a estudiantes se está sintiendo tan estancado/a, como para que no sientan ánimo de continuar.

## Actividad de Cierre (5 - 10 minutos)

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que han aprendido, por qué es útil, y cómo se sienten puede ayudar a solidificar cualquier conocimiento obtenido hoy y construir una hoja de revisión que pueden revisar en el futuro.

Sugerencias para el Diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- ¿Qué tipo de errores encontraste hoy?
- Dibuja un error que hayas encontrado en uno de los desafíos hoy. ¿Qué hiciste para "debug" el programa?

## Aprendizaje Extendido

Utilice estas actividades para mejorar el aprendizaje de los/as estudiantes. Pueden ser utilizados como actividades fuera de la clase u otro enriquecimiento.

### Plantar bugs

Pida a los/as estudiantes que vuelvan a los niveles anteriores, añadiendo errores a sus soluciones. A continuación, pueden pedir a otros/as estudiantes que apliquen debug a su trabajo. Esto también se puede hacer con desafíos de papel.

Cuando otros/as estudiantes estén debugging, asegúrese de que las críticas sean constructivas. Si esto fuera un problema para su clase, repase acerca del debugging respetuoso antes de esta actividad jugando con otro/a estudiante.

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

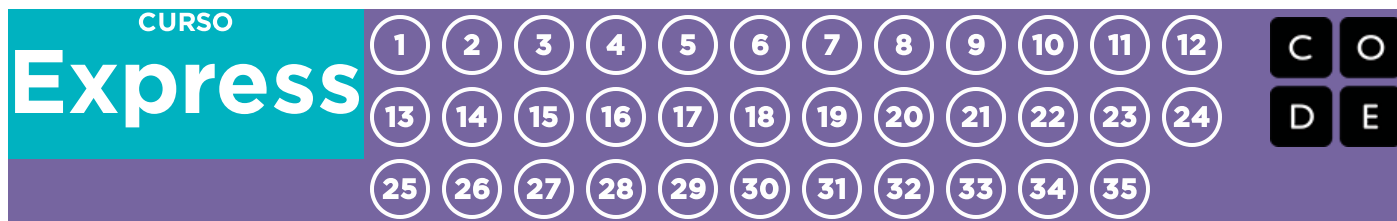
- **AP** - Algoritmos y programación



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



# Lección 5: Bucles: mis amigos robóticos

Sin conexión | Bucle | Repetir

## Descripción

Esta lección se basa en la lección ["Mis Amigos Robóticos" de[0]. Aquí, los estudiantes aprenden la simplicidad y utilidad de los loops “programando” a sus amigos usando el idioma de "Mis Amigos Robóticos". Una vez que se introduzcan los loops, los estudiantes encontrarán que pueden construir estructuras más grandes de manera más rápida.

## Propósito

Esta lección es una introducción a los bucles. A través de los *bucles*, los estudiantes pueden agrupar comandos que necesitan repetirse y, así, simplificar sus códigos. Al notar la repetición en los movimientos de sus compañeros y determinar cuántas veces se deben repetir sus códigos, los estudiantes desarrollarán habilidades de pensamiento crítico.

## Agenda

### Actividad previa (10 - 15 min)

Repaso de Mis amigos robóticos  
Bucles al rescate

### Actividad Principal (15 - 20 min)

Haciendo loops con Mis Amigos Robóticos

### Actividad de Cierre (8 min)

Registrar en un diario

### Actividades de Extensión

## Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Identificar código repetitivo y convertir una serie de acciones en un solo bucle.
- Decodificar bucles en una serie de acciones.

## Preparación

- ☐ Mire el **Mis Amigos Robóticos Loopy - Video para el docente**.
- ☐ Imprima un **Loops Mis Amigos Robóticos - Guía de preparación para el docente** para cada grupo.
- ☐ Consiga hasta 20 vasos de papel para cada grupo.
- ☐ Asegúrese de que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones**.

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Mis Amigos Robóticos Loopy** - Video para el docente
- **Programación: Mis Amigos Robóticos** - Guía de preparación para el docente
- **Loops Mis Amigos Robóticos** - Guía de preparación para el docente
- **Mis Amigos Robóticos** - Video para el docente

Para los estudiantes

- **Diario Think Spot** - Diario de reflexiones

Hacer una copia ▾

## Vocabulario

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



# Guía docente

## Actividad previa (10 - 15 min)

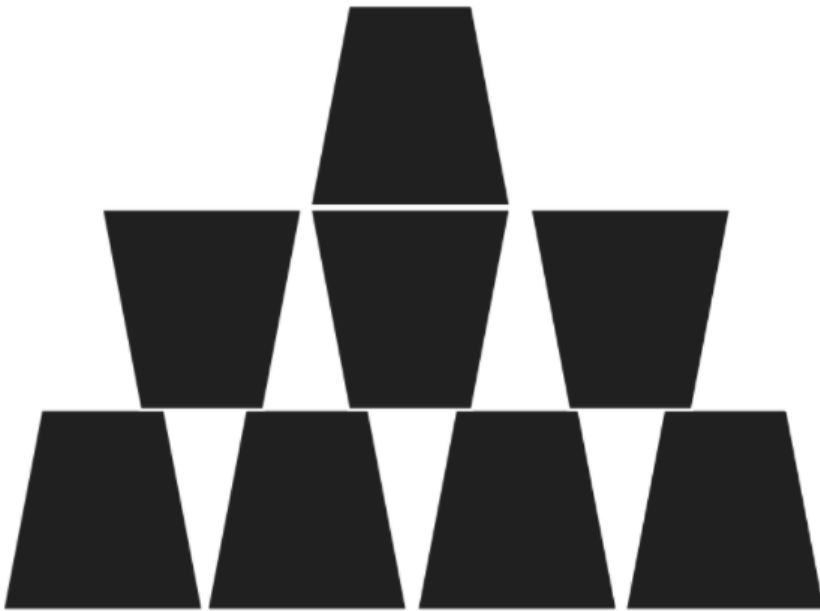
### Repaso de Mis amigos robóticos

Objetivo: con este repaso, los estudiantes recordarán que los programas de la actividad Mis amigos robóticos se pueden volver más complejos rápidamente.

Explica las reglas de Mis amigos robóticos.

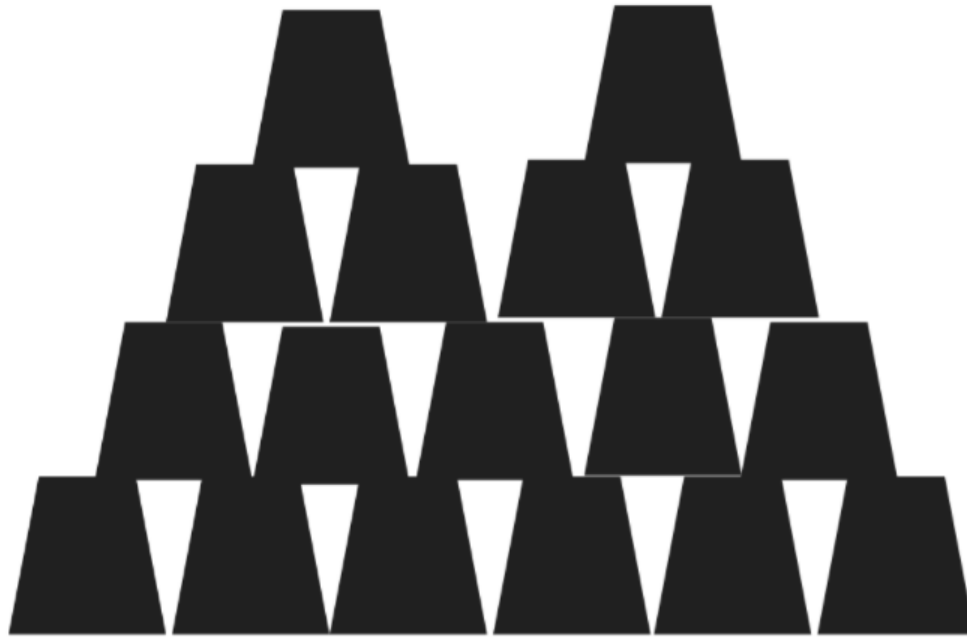
Con la clase como grupo, muestra un desafío sencillo del Paquete de vasos apilados de Mis amigos robóticos y programa con todos como un recordatorio de las reglas y la terminología.

Luego, presenta un desafío un poco más difícil, pero que también requiera muchos pasos como el siguiente.



Pide a un voluntario (o grupo de voluntarios) que pase al frente para ayudar a programarlo en la pizarra. Si haces que sigan estrictamente la regla de "no usar más símbolos que los que están en la clave", ¡probablemente tardarán un poco más!

Ahora, muestra esta imagen:



¿Cuál es la reacción de la clase?

## Bucles al rescate

Brinda a los alumnos la oportunidad de pensar ideas sobre formas más cortas de transmitir el código que están a punto de crear (esta parte se puede omitir si tus estudiantes comienzan a decir cosas como: "Avanzar 6 veces", debido a que abrirá el debate acerca de cómo mostrar "seis veces" con símbolos).

Una vez que los estudiantes han comprendido la idea de "repetir" código, preséntales el vocabulario relacionado. Asegúrate de compartir con ellos que, a menudo, los términos "repetir algo" y "enlazar algo" tienen el mismo significado en Code.org.

## Actividad Principal (15 - 20 min)

### Haciendo loops con Mis Amigos Robóticos

Objetivo: Esta actividad le dará a los/as estudiantes la posibilidad de adquirir práctica localizando áreas donde los loops de repetición se pueden usar, así como lugares donde necesitan expandir los programas que utilizan loops.

La práctica hace al Maestro

Tome el programa de uno de sus pilas de vasos previos y muéstrela a sus estudiantes. Pídale que lo/a ayuden a encontrar los lugares dónde se repite la misma flecha, sin interrumpir, múltiples veces. Pídale a los/as estudiantes que cuenten la cantidad de veces que esas flechas se repiten y que le den el recuento final.

Encierre en un círculo la primera flecha en esa línea, escriba el número de loops cerca de ese círculo, luego haga una cruz sobre el resto de las flechas.

Repita esto hasta que el programa completo haya sido acortado, luego re escriba el programa de una forma que los/as estudiantes puedan ver cuánto más simple son las instrucciones resultantes.

Haciendo loops con Mis Amigos Robóticos

Ahora que los/as estudiantes tienen una nueva herramienta en su caja de herramientas, podrían comenzar a encontrar el éxito en una nueva (y más difícil) pila de vasos.

Ponga a los/as estudiantes a trabajar con las tarjetas del Paquete más difícil de Loops Mis Amigos Robóticos y vea cómo lo hacen. Puede continuar trabajando con todos/as juntos o separar a los/as estudiantes en pequeños grupos, lo que sea mejor para su aula.

## Actividad de Cierre (8 min)

### Registrar en un diario

Objetivo: hacer que los estudiantes reflexionen sobre la actividad que acaban de experimentar.

Conversación rápida:

Estos son algunos temas posibles:

- ¿Sientes que los bucles hacen que la programación sea más fácil o más difícil?
- ¿Qué otros tipos de cosas de la vida repetimos?
  - Comer: llevarse la comida a la boca, masticar 20 veces
  - Cepillarse el pelo: cepillar el pelo 35 veces
  - Rutinas: despertarse, ir a la escuela, volver a casa, acostarse

Sugerencias para el diario:

- ¡Es tiempo de ocuparnos del diario! Pide a los estudiantes que dibujen una cara en la esquina de la página de su diario que les recuerde cómo se sintieron en esta lección.
- Haz que los estudiantes escriban o dibujen algo en su diario para recordar más adelante qué son los bucles. Esto puede provenir de una idea como:
  - ¿Qué significa "repetir" para ti?
  - Haz un dibujo de ti repitiendo algo.

### Teaching Tips:

Asegúrese de estar atento/a cuando los/as estudiantes estén usando loops. Evite corregir sus algoritmos generales, pero sienta la libertad de señalar las instrucciones que pueden ser acortadas usando un ciclo de repetición.

Observe a los/as estudiantes a medida que recorren el código. ¿Hay algún bug? Use las preguntas de debugging para ayudarlos/as a encontrar una solución.

- ¿Qué hace?
- ¿Qué se supone que debe hacer?
- ¿Qué te dice eso?
- ¿Funciona en el primer paso?
- ¿Funciona en el segundo paso?
- ¿Dónde deja de funcionar?

## Actividades de Extensión

- Pida a sus alumnos que dibujen sus propias pilas de vasos para que alguien las modifique.
- Proporcione a los alumnos algoritmos que utilicen repeticiones, luego pídales que amplíen el programa a una versión completa paso a paso.

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

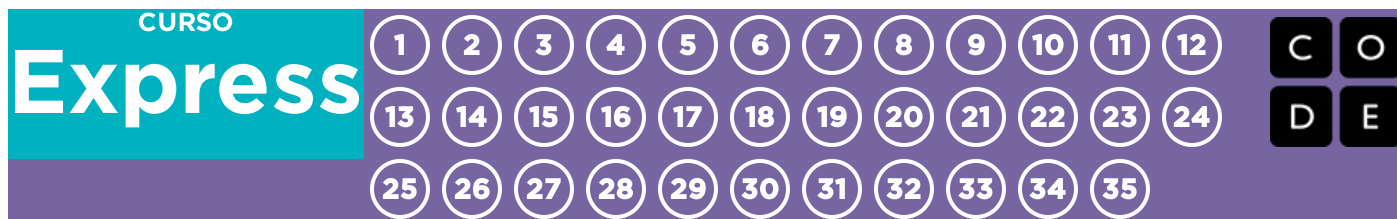
- AP - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



# Lección 6: Bucles en Artista

Bucle | Artista

## Descripción

¡Mira a la cara de los/as estudiantes mientras hacen sus propios diseños maravillosos usando un pequeño número de bloques y pegatinas digitales! Esta lección se basa en la comprensión de loops de lecciones anteriores y da a los/as estudiantes la oportunidad de ser verdaderamente creativos. Esta actividad es fantástica para producir artefactos para portafolios o para reuniones de padres/profesores.

## Propósito

Esta serie destaca el poder de los loops con diseños creativos y personales.

Ofrecido como una secuencia respaldada por el proyecto, esta progresión permitirá a los/as estudiantes construir sobre su propio trabajo y crear artefactos asombrosos.

## Agenda

### Actividad previa (15 mín)

Introducción

### Actividad Puente - Bucles (15 mín)

Actividad sin conexión con bloques de papel  
Previsualizar desafíos en línea como una clase

### Actividad Principal (30 mín)

Curso Rápido Fundamentos de Ciencias de la Computación  
- Sitio web

### Actividad de Cierre (15 mín)

Escribir en el Diario

## Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Identificar los beneficios de usar una estructura de bucles en lugar de la repetición manual.
- Diferenciar entre comandos que deben repetirse en bucles y comandos que deben usarse solos.

## Preparación

- ▢ Juegue **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web** correspondiente a este curso para encontrar áreas de potenciales problemas para su clase.
- ▢ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones.**
- ▢ Asegúrese que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones.**

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones** [Hacer una copia ▼](#)

Para los estudiantes

- **Bloques Blockly Sin Conexión (Cursos 2-5)**  
- Manipuladores
- **Diario Think Spot - Diario de reflexiones**  
[Hacer una copia ▼](#)

# Vocabulario

- **Repetir** - hacer algo de nuevo.
- **Bucles mientras** - un bucle que se sigue repitiendo mientras una condición es verdadera.

# Guía docente

## Actividad previa (15 mín)

### Introducción

Los estudiantes ya deberían conocer los bucles en este punto. Basado en lo que piensas que podría beneficiar a tu clase, recomendamos:

- Crear un nuevo baile con bucles como en "Getting Loopy"
- Como clase, recorrer un desafío de la última lección, "Bucles en el Laberinto"
- Repasar el uso del Artista mediante un desafío de "Programación en el Artista"
- Previsualizar un desafío de esta lección

Todas estas opciones permiten repasar los bucles o el artista, ¡lo que ayudará a preparar a tu clase para divertirse con los desafíos en línea!

## Actividad Puente - Bucles (15 mín)

Esta actividad ayudará a llevar los conceptos sin conexión de "Mis locos amigos robóticos" al mundo en línea en el que se mueven los estudiantes. Elige *una* de las siguientes opciones para hacer con tu clase:

### Actividad sin conexión con bloques de papel

Selecciona un patrón de **Apilando Ideas - Manipuladores** de la actividad sin conexión Mis amigos robóticos y asigna a los estudiantes los bloques **Bloques Blockly Sin Conexión (Cursos 2-5) - Manipuladores** que ya incluyen el bucle `repetir` y los comandos `mover`\_\_. Luego, haz que los estudiantes programen un "robot" (un compañero o el docente) desde sus escritorios para obtener la pila correcta de vasos. ¡Asegúrate de que entiendan que los bloques deben ir de arriba hacia abajo y que todos deben tocarse! Haz que cada pareja de estudiantes comparta sus respuestas y resuelva las preguntas o errores que puedan surgir.

### Previsualizar desafíos en línea como una clase

Toma un desafío de los desafíos en línea correspondientes en **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**. Mediante **Programación: Mis Amigos Robóticos - Clave de símbolo** de la actividad sin conexión Mis locos amigos robóticos, haz que los estudiantes dibujen un patrón que piensan que puede hacer el Artista. Pide a los estudiantes que lo compartan. ¡Mira cuántos estudiantes tienen la misma respuesta!

## Actividad Principal (30 min)

### Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

Algunos estudiantes pueden descubrir dónde agregar bucles de "repetición" escribiendo el programa sin los bucles y luego encerrando en un círculo las secciones de repeticiones. Si crees que los estudiantes de tu clase pudiesen beneficiarse de esto, pídeles que tengan consigo papel y lápices. Los estudiantes también pueden disfrutar dibujando algunas de las formas y figuras en papel antes de programarlo en línea. (Al dibujar estampillas, puede ser más fácil simbolizar aquellos con formas simples como círculos y cuadrados)

## Actividad de Cierre (15 mín)

### Escribir en el Diario

Hacer que los alumnos escriban acerca de lo que han aprendido, por qué es útil, y cómo se sienten puede ayudar a solidificar cualquier conocimiento obtenido hoy y construir una hoja de revisión que pueden revisar en el futuro.

#### Sugerencias para el Diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- ¿Cuál fue la forma o figura más genial que programaste hoy? ¡Dibújala!
- ¿Cuál otra forma o figura te gustaría programar? ¿Puedes crear el código para crearla?

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

► **AP** - Algoritmos y programación



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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

# Lección 7: Loops Anidados

Loops Anidados | Bucles

## Descripción

En esta actividad online, los/as estudiantes tendrán la oportunidad de comprender acerca de los loops a un nivel completamente nuevo. Jugando con Bee y las Plantas vs Zombies, los/as estudiantes aprenderán a programar un loop dentro de otro loop. También se les animará a averiguar cómo pequeños cambios en cada loop afectarán su programa cuando hagan clic en Ejecutar .

## Propósito

En esta introducción a los loops anidados, los/as estudiantes saldrán de su zona de confort para crear soluciones más eficientes para los desafíos.

En los desafíos anteriores, los loops empujaban a los/as estudiantes a reconocer la repetición. Aquí, los estudiantes aprenderán a reconocer patrones dentro de patrones repetidos para desarrollar estos loops anidados. Esta etapa comienza al animar a los/as estudiantes a que intenten resolver un enigma donde el código es irritante y complejo para escribir. Después de que un video presenta los loops anidados, se muestra a los/as estudiantes un ejemplo y se les pide que predigan lo que sucederá cuando se coloque un loop dentro de otro loop. Esta progresión lleva a mucha práctica para que los/as estudiantes se solidifiquen y desarrollen su comprensión de los loops en la programación. .

## Agenda

### Actividad previa (10 min)

Introducción

### Actividad Principal (30 min)

Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

### Actividad de Cierre (15 mín)

Escribir en el Diario

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Dividir las tareas complejas en secciones repetibles más pequeñas.
- Reconocer patrones grandes repetidos conformados de patrones repetidos más pequeños.
- Identificar los beneficios de usar una estructura de bucles en lugar de la repetición manual.

## Preparación

- ▣ Juegue **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web** para encontrar áreas de potenciales problemas para su clase.
- ▣ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones.**
- ▣ Asegúrese que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones.**

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones** Hacer una copia ▾

Para los estudiantes

- **Diario Think Spot - Diario de reflexiones** Hacer una copia ▾

## Vocabulario



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

# Guía docente

## Actividad previa (10 min)

### Introducción

Repase brevemente con la clase qué son los loops y por qué los usamos.

- ¿Qué hacen los loops? - Los loops repiten un conjunto de comandos. (ver vocabulario en comando si los/as estudiantes no lo reconocen)
- ¿Cómo utilizamos los loops? - Utilizamos loops para crear un patrón hecho de acciones repetidas.

Dígale a la clase que ahora van a hacer algo súper genial: usar loops dentro de los loops. Pida a la clase que prediga para qué tipo de cosas usaríamos un loop dentro de un loop.

"Si un loop repite un patrón, entonces un loop repetirá un patrón de patrones".

Los/as estudiantes no necesitan entender esto de inmediato, así que siéntase libre de pasar a los desafíos online incluso si los/as estudiantes todavía parecen un poco confundidos. .

## Actividad Principal (30 min)

### Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

Recomendamos fervientemente **Programación en Parejas - Video para el estudiante** en esta lección. Puede que este no sea un tema sencillo para la mayoría de los estudiantes. Trabajar con un compañero y debatir posible soluciones para los desafíos podría brindar tranquilidad a los estudiantes.

Además, asegúrate de que haya papel y lápiz para que los estudiantes escriban su plan antes de escribir el código. Algunos desafíos tienen un límite sobre la cantidad de ciertos bloques que puedes usar, de modo que si los estudiantes desean escribir la respuesta completa para encontrar las repeticiones, el papel les puede ser útil.

## Actividad de Cierre (15 mín)

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil y cómo se sienten, puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el Diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- ¿Qué es un loop anidado?
- ¿Puedes dibujar un desafío que utilice un loop anidado? Intenta codificar la solución a tu propio desafío.

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

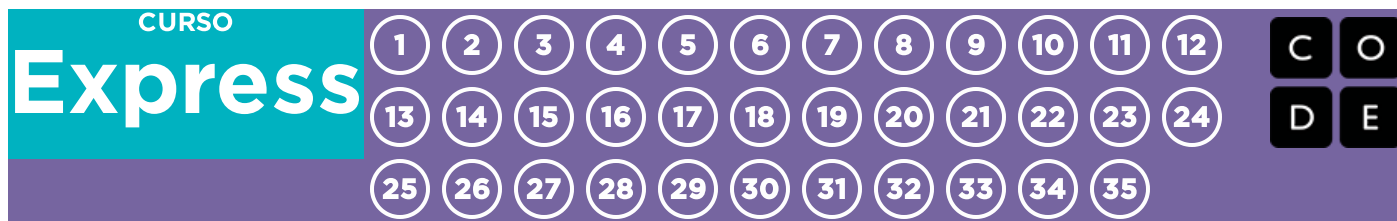
- **AP** - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



# Lección 8: Bucles anidados en Frozen

Bucle | Bucle anidado | Artista

## Descripción

Ahora que los/as estudiantes saben cómo hacer sus loops, pueden crear muchas cosas hermosas. Esta lección llevará a los/as estudiantes a través de una serie de ejercicios para ayudarles a crear su propio portafolio de imágenes ¡usando las excelentes habilidades de patinaje en el hielo de Anna y Elsa!

## Propósito

En esta serie, los/as estudiantes practicarán anidando loops mientras crean imágenes que estarán emocionados de compartir.

Comenzando con algunas instrucciones, los/as estudiantes harán sus propias decisiones en lo que respecta a la creación de diseños para la repetición. Luego las girarán de diferentes maneras para terminar con una obra de arte verdaderamente única.

## Agenda

### Actividad previa (15 mín)

Introducción

### Actividad Principal (30 mín)

Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

### Actividad de Cierre (15 mín)

Escribir en el Diario

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Definir cuándo se necesita un bucle o un bucle anidado, o cuándo no se necesita ningún bucle.
- Reconocer la diferencia entre usar un bucle y un bucle anidado.
- Separar el código en secuencias repetibles más grandes mediante bucles y bucles anidados.

## Preparación

☐ Juegue **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web** correspondiente a esta lección para encontrar cualquier potencial área problemática para su clase.

☐ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones**.

☐ Asegúrese de cada estudiante tenga un **Diario Think Spot - Diario de reflexiones**.

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones** [Hacer una copia ▼](#)

Para los estudiantes

- **Diario Think Spot - Diario de reflexiones** [Hacer una copia ▼](#)

# Vocabulario

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

# Guía docente

## Actividad previa (15 mín)

### Introducción

Pida a sus estudiantes que discutan sobre el último grupo de desafíos.

- ¿Qué les gustó o no les gustó?
- ¿Qué desafíos fueron difíciles? ¿Por qué?
- ¿Qué desafíos fueron fáciles? ¿Por qué?
- Si fueras a enseñar los loops anidados a un/a amigo/a, ¿qué dirías para ayudarlos a entender?

Si hay tiempo, de una introducción a los personajes principales de los desafíos de hoy, Anna y Elsa de Frozen. Cuénteles la historia de las hermanas si es que la clase no la sabe. Para crear emoción, cuénteles que ¡usarán loops anidados para hacer algunos dibujos fantásticos con los patines de hielo de Anna y Elsa!

## Actividad Principal (30 min)

### Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

Este conjunto de desafíos está pensado como una progresión. Esto significa que cada desafío sienta una base para el siguiente. Los estudiantes harán cambios pequeños y simples en el código que ya han escrito para que los diseños sean cada vez más interesantes.

## Actividad de Cierre (15 mín)

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil y cómo se sienten, puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el diario:

- ¿De qué trató la lección de hoy?
- ¿De qué trató la lección de hoy?
- ¿Cuándo usas un loop? ¿Cuándo usas un loop anidado?
- Ejercicio para pensar: ¿Puedes hacer de todo un loop anidado? ¿Puedes sólo con un loop normal? ¿Puedes dibujar un ejemplo?
  - Respuesta: Si, puedes, pero es mucho más difícil. Los loops anidados hacen los programas más simples.

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

- AP - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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

# Lección 9: Más allá de la Programación: Internet

Sin conexión | Internet

## Descripción

A pesar de que muchas personas usan Internet diariamente, no muchos saben cómo funciona. En esta lección, los estudiantes harán como que navegan a través de Internet, aprendiendo sobre conexiones, URLs, direcciones IP y DNS.

## Propósito

Si ha estado haciendo cada lección en este curso, entonces cada estudiante de su clase ha utilizado Internet... pero ¿cuántos saben cómo funciona? Aprender más sobre Internet ayudará a los estudiantes a desarrollar una mejor comprensión de sus infinitas posibilidades.

## Agenda

### Actividad previa (20 min)

Vocabulario

Comprender el mensaje

### Actividad Principal (20 min)

Internet

### Actividad de Cierre (15 min)

Charla Exprés: ¿Qué aprendimos?

### Evaluación (5 min)

Internet - Evaluación

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Aprender sobre la complejidad de enviar mensajes a través de Internet.
- Traducir URL a direcciones IP.

## Preparación

- ☐ Vea el **Internet - Video para el docente**.
- ☐ Imprima suficientes **Tarjetas de dirección IP y tarjetas de tipo entrega - Manipuladores** para cada grupo.
- ☐ Imprima un **Internet - Evaluación** para cada estudiante.
- ☐ Entre a Internet (por ejemplo a **get-site-ip.com**).
- ☐ Asegúrese de que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones**.

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Internet** - Video para el docente
- **Internet** - Guía de respuestas de la evaluación [Hacer una copia](#)

Para los estudiantes

- **Tarjetas de dirección IP y tarjetas de tipo entrega - Manipuladores** [Hacer una copia](#)
- **Internet - Evaluación** [Hacer una copia](#)

## Vocabulario

- **DNS** - abreviatura de Domain Name System (sistema de nombres de dominio),

que traduce nombres de dominio (como ejemplo.com) a direcciones IP (como 93.184.216.34).

- **DSL/Cable** - un método para enviar información mediante cables telefónicos o de televisión.
- **Cable de fibra óptica** - una conexión que utiliza la luz para transmitir información.
- **Internet** - un grupo de computadoras y servidores que están conectados entre sí.
- **Dirección IP** - un número asignado a cualquier elemento que esté conectado a Internet.
- **Paquetes** - pequeños fragmentos de información que se formaron minuciosamente a partir de fragmentos más grandes.
- **Servidores** - computadoras cuya función es proporcionar cosas a otros.
- **URL** - una dirección fácil de recordar que llama a una página web (como [www.code.org](http://www.code.org)).
- **Wi-Fi** - un método inalámbrico para enviar información mediante ondas de radio.



# Guía docente

## Actividad previa (20 min)

### Vocabulario

Esta lección tiene varias palabras nuevas e importantes:

- **Dirección IP** - Dilo conmigo: Di-rec-cción I-P

Un número asignado a cualquier ítem que esté conectado a Internet

- **DNS (Servidor de nombre de dominio)** - Dilo conmigo: D-N-S

El servicio que traduce URLs a direcciones IP

- **URL (Localizador universal de recursos)** - Dilo conmigo: U-R-L

Una dirección fácil de recordar para llamar a una página web (como [www.code.org](http://www.code.org))

- **Internet** - Dilo conmigo: In-ter-net

Un grupo de computadoras y servidores que están conectados entre sí

- **Servidores** - Dilo conmigo: Ser-vi-do-res

Computadoras que existen solo para brindarles cosas a otras

- **Cable de fibra óptica** - Dilo conmigo: Ca-ble de fi-bra óp-ti-ca

Una conexión que usa luz para transmitir información

- **Wi-fi** - Dilo conmigo: Wi-Fi

Un método inalámbrico de envío de información que usa ondas de radio

- **DSL/Cable** - Dilo conmigo: D-S-L/Ca-ble

Un método de envío de información que usa cables de teléfono o televisión

- **Paquetes** - Dilo conmigo: Pa-que-tes

Pequeñas porciones de información que se han formado cuidadosamente a partir de porciones más grande de información

### Comprender el mensaje

- Es muy probable que tus estudiantes sepan lo que es Internet, pero puede que no entiendan realmente lo que hace Internet.

- Pregunta "¿Qué es la Internet?"
  - ¿La Internet es un espacio público o un espacio privado?
  - (La verdad es que mucha gente cree que puede ser ambas cosas, pero debe ser vista como un espacio público sin importar que ámbitos hayas logrado dominar).
  - ¿Cómo hace la información para ir de un lado a otro?
- Digamos que quiero mirar la página web de Code.org. ¿Cuál crees que sería el proceso para que mande un mensaje para solicitar esa página?
    - ¿Qué tengo que hacer como usuario?
    - ¿Qué crees que sucede dentro de Internet?

#### 💡 Sugerencia para la lección

Lo único que necesitas es una rápida vista previa. Todas estas palabras serán explicadas como parte de la lección, de modo que sería mucho más confuso hacer una breve introducción de las palabras como un tipo de indicación al estilo "fíjate si puedes detectar estas palabras durante el día".

#### 💡 Sugerencia para la lección

Hay algunos videos muy buenos de YouTube sobre este tema que pueden hacer que esta lección sea más fácil de comprender. Se los puedes mostrar a la clase con antelación, o simplemente puedes mirarlos tú mismo. **Esta es una de las versiones más claras y entretenidas.** (Recomendamos parar el video en el minuto 2:59, si es posible).

Enviar un mensaje por Internet es muy parecido a enviar un mensaje por correo postal... ¡si cada carta que enviáramos requiriese miles de sobres!

Cada mensaje que enviamos a través de Internet se corta en pedazos y cada pieza se envuelve en su propia versión de sobre. Esos recibien del nombre de "paquetes". Los paquetes están formados especialmente por porciones de información que pueden fluir fácilmente por cualquier canal de Internet.

A veces, unos pocos de esos paquetes se perderán, porque Internet es un lugar alocado. En ese caso, los paquetes deben reenviarse, y todo el mensaje se tiene que poner en pausa hasta que lleguen.

¿A dónde crees que se dirigen esos paquetes?

- Incluso si estás enviando mensajes a otra persona, primero tienen que llegar, al menos, a un "servidor".
  - Un servidor es una computadora especial que se supone que siempre debe estar encendida y lista para enviar y recibir información.
  - Todo sitio web tiene un servidor.
  - Incluso el correo electrónico va a servidores.

Los servidores no tienen nombres como tú y yo. Se hace referencia a ellos usando números. Estos números se llaman direcciones IP, y tienen un aspecto un poco extraño.

- Por ejemplo: Una de las direcciones IP de Code.org solía ser 54.243.71.82
  - (Asegúrate de mirar esto con antelación. La mayoría de las direcciones IP cambian de tanto en tanto, y luego se reutilizan para otros sitios).

Hay muchas formas de llegar a Internet desde tu hogar, escuela o lugar de trabajo.

- Puedes conectarte directamente usando un cable (que podría ser DSL, Cable o Fibra Óptica)
- O puedes conectarte usando ondas de radio a través del aire mediante Wi-Fi

Las conexiones directas son más confiables, pero pueden ser incómodas.

- ¿Puedes descubrir por qué?
  - (¡Tienes que estar conectado a un cable!)

Las conexiones de Wi-Fi son muy cómodas, pero no siempre son seguras.

- ¿Puedes descubrir por qué no?
  - (Las ondas de radio rebotan en todo el lugar y se pueden perder).

De modo que si estás acostumbrado a enviar información a URLs (como **www.code.org**) y los servidores de hecho tienen direcciones IP como nombres (como 54.243.71.82), ¿cómo hace Internet para cambiar de una a la otra? Esa es la función de un DNS. El DNS (Servidor de nombre de dominio) tiene tablas que le permiten al sistema ir de una a otra entre URLs y direcciones IP. Si alguna vez los Servidores de nombre de dominio dejan de funcionar, apagaría la Internet tal y como la conocemos.

Dicho lo cual, intentemos comprender qué hace el DNS haciendo nosotros mismos un pequeño DNS.

Toma un pedazo de papel y dibuja una grilla similar a la de la actividad de Internet:

Ejemplo de tabla de DNS:

| # | URL      | Dirección IP |
|---|----------|--------------|
| 1 | code.org | 54.243.71.82 |
| 2 |          |              |

#### 💡 Sugerencia para la lección

Si crees que esto es demasiado texto y sería muy aburrido ilustrarlo ante una clase llena de niños de escuela primaria, ¡tienes toda la razón! Si puedes mostrar un video de YouTube en clase para ayudar a explicarlo todo, recomendando fervientemente dibujar imágenes para explicar cada una de las ideas anteriores, o elegir estudiantes como voluntarios para representar lo que describes mientras lo estás explicando. No se supone que deban entender cada detalle y definición en este momento, solo ganar exposición.

| # | URL | Dirección IP |
|---|-----|--------------|
| 3 |     |              |
| 4 |     |              |
| 5 |     |              |

Primero, debemos completar esta tabla.

- Haz una encuesta en la clase sobre sus sitios web favoritos y escribe las URLs en la columna de la izquierda
- Usa un sitio como **get-site-ip.com** para encontrar las direcciones IP para esos sitios y escríbelas en las filas correspondientes de la columna derecha.

Ahora tomemos esta tabla de DNS y haz de cuenta que envías mensajes por Internet.

## Actividad Principal (20 min)

### Internet

#### Instrucciones:

- Cree sus propias tablas DNS, similar a la que se muestra más arriba.
- Pida a la clase que lo/a ayude a completar los espacios vacíos de la tabla. Escoja sus URLs favoritas y encuentre sus direcciones IP usando un sitio como **www.get-site-ip.com**. Nota: El sitio web [www.get-site-ip.com](http://www.get-site-ip.com) tiene la opción de traducir al español su contenido, para ello, debes ingresar desde el buscador Google a este sitio web, y luego hacer clic a la tecla derecha de tu mouse/touchpad y seleccionar la opción “traducir al español”, de esta forma podrás visualizar todo el contenido en español.
- Divídalos en grupos de 3 a 5.
- Asigne a cada grupo una dirección IP de la tabla recién creada y asigne a cada persona del grupo una posición.
  - El/a Emisor/a del Mensaje.
  - Internet.
  - El Servidor (lleva las direcciones IP).
  - Internet Retorno (opcional).
  - El/a Receptor/a del Mensaje (opcional).
- Cada grupo dibujará una **Tarjetas de dirección IP y tarjetas de tipo entrega - Manipuladores** para averiguar dónde va su mensaje y cuál será el método de entrega del mensaje (Wi-Fi, Cable/DSL o Cable de Fibra Óptica).
- El/a Emisor/a del Mensaje creará una nota para enviar al servidor.
- Internet cortará el papel en cuatro partes pequeñas llamadas paquetes, y luego entregará un paquete a la vez al Servidor con la dirección IP que se sacó del mazo de Tarjetas.
- El Servidor se asegura que el mensaje llegue en orden, luego enviará cada paquete de a uno con el Internet Retorno (puede ser la misma persona u otra diferente al Internet original).
- Internet Retorno entregará cada pieza de vuelta al Receptor/a del Mensaje (puede ser la misma persona u otra diferente al/a Receptor/a del Mensaje) y lo une nuevamente.
- El/a Receptor/a del Mensaje esperará que lleguen todas las piezas, ¡luego leerá el mensaje para asegurarse que llegó de forma correcta!

#### Reglas:

- Internet debe cortar el mensaje en 4 paquetes exactos.
- Si Internet deja caer un paquete, lo deben recoger y volver al comienzo para entregarlo nuevamente.
- El Servidor debe esperar que lleguen todas las partes del mensaje antes de comenzar a enviarlo.

#### Información:

- Wi-Fi: Conveniente, pero irregular. El Wi-Fi no requiere cables, sin embargo, debido a que la señal rebota por todos lados, los paquetes se pueden perder fácilmente.

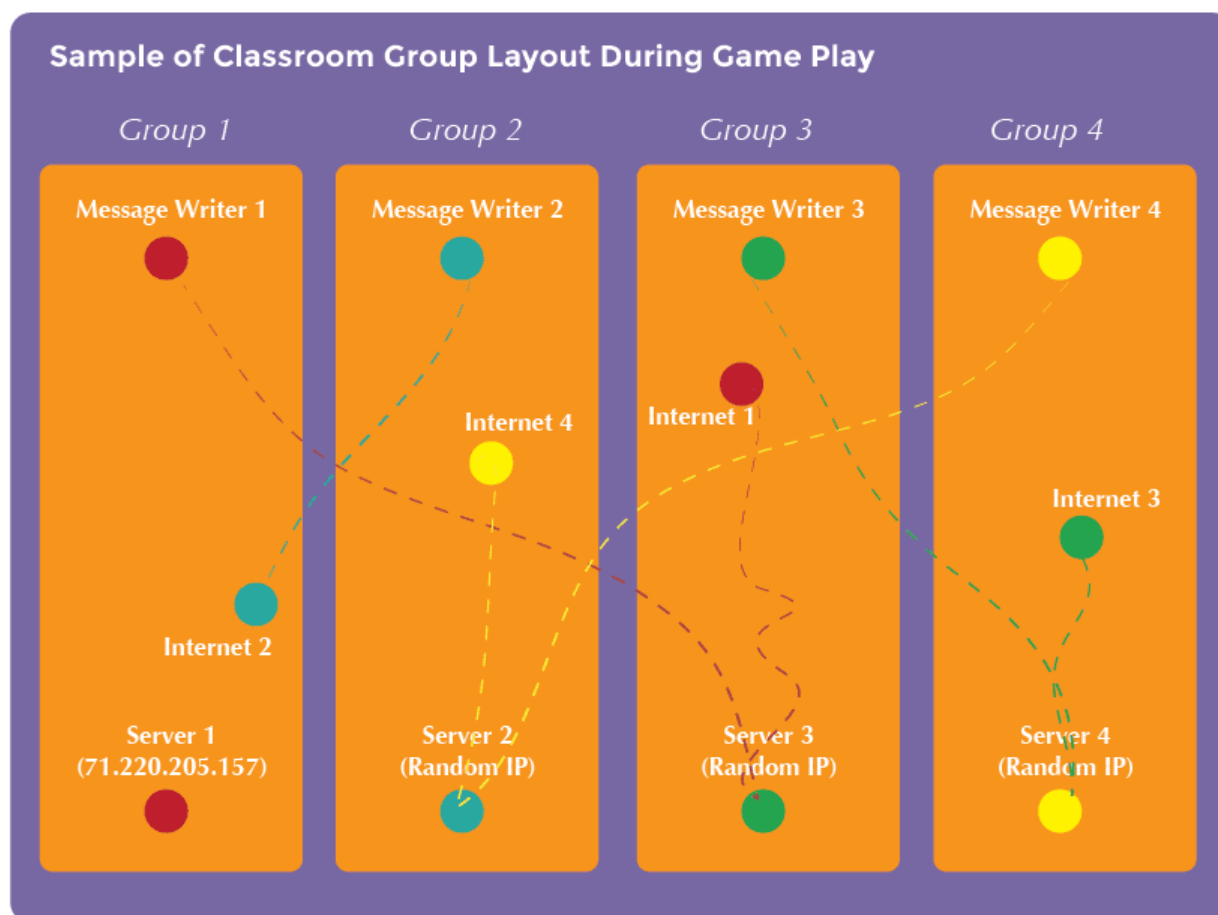
- Simulación Internet debe llevar cada paquete en los hombros (no en las manos).
- Cable/DSL: Bastante Buena para entregar mensajes, pero debe estar conectado a un cable.
  - Simulación Internet debe llevar cada paquete en el dorso de la mano y con la otra mano debe estar tocando siempre un muro, escritorio, silla el suelo.
- Cable de Fibra Óptica: El mejor para entregar mensajes, pero debe estar conectado a un cable.
  - Simulación: Internet puede llevar los paquetes en la mano, pero debe mantener la otra mano tocando siempre un muro, escritorio, silla o en el piso.

Para hacer este juego, puede ubicar a los grupos reunidos en cualquier lugar, pero la primera vez puede ser menos confuso hacer que los grupos jueguen en filas.

- Ponga en fila a los “Servidores” en un extremo de la sala (sosteniendo sus direcciones IP). Los/as jugadores/as Internet Retorno pueden también estar ahí (si tiene esa cantidad de gente en cada grupo).
- Ponga la fila con todos los otros frente a su servidor al otro lado de la sala.
- Es probable que los emisores de mensaje envíen sus mensajes a un servidor distinto al propio, así que es probable que los jugadores de Internet pasen de un grupo a otro. Se puede ver similar al dibujo más abajo (en Inglés):

#### 💡 Recomendación para la lección

Si siente que hay demasiadas reglas que deben ser explicadas, siéntase libre de ponerlas en la pizarra y explicarlas únicamente a medida que avanza el juego. Puede jugar múltiples rondas hasta que la clase entienda completamente.



## Actividad de Cierre (15 mín)

Charla Exprés: ¿Qué aprendimos?

?

## Evaluación (5 min)

### Internet - Evaluación

Distribuya la hoja de la evaluación y permita que los/as estudiantes completen la actividad en forma independiente luego que las instrucciones queden claras. Esto debería serles familiar debido a las actividades en lecciones previas.

#### 💡 Lesson Tip

Las preguntas de la Charla Exprés tienen como objetivo incentivar el pensamiento a gran escala pensando en cómo la lección se relaciona con el gran mundo de los/as estudiantes y su incluso, más grande futuro. Utilice el conocimiento de su aula para decidir si quiere discutir las como clase, en grupos pequeños o compañeros/as de banco.

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

► **NI** - Redes y la Internet



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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

# Lección 10: Ciudadanía Digital: Practicando Ciudadanía Digital

Common Sense Education | Sin conexión

## Descripción

En colaboración con el Sitio Web de Common Sense Education, esta lección ayuda a los/as estudiantes a aprender a pensar, en forma crítica, sobre la información de usuario/a que algunos sitios webs solicitan o requieren. Los/as estudiantes aprenden la diferencia entre información privada e información personal, distinguiendo lo que es seguro e inseguro de compartir online. Los/as estudiantes también exploran lo que significa ser responsables y respetuosos con sus comunidades offline y online, como un paso hacia el aprendizaje de cómo ser buenos ciudadanos digitales.

## Propósito

A medida que los/as estudiantes pasan más tiempo en los computadores, deberían estar conscientes que el internet no siempre es un espacio seguro. En esta lección, a los/as estudiantes se les enseña cuál información es segura compartir y qué información debería permanecer privada. Los/as estudiantes crearán “superhéroes” y aprenderán lo que significa ser un Ciudadano Digital en internet.

## Agenda

### Actividad previa (15 mín)

Vocabulario

Online Personal vs. Privado

### Actividad Principal (30 - 40 mín)

Plantillas De Superhéroe de Cubecraft - Manipuladores

### Actividad de Cierre (15 mín)

Conversación Rápida: ¿Qué aprendimos?

Escribir en el Diario

### Evaluación (5 mín)

Ciudadanía Digital - Evaluación

### Aprendizaje Extendido

## Ver en Code Studio

## Objetivos

### Los estudiantes podrán:

- Comparar y contrastar las responsabilidades con las comunidades conectadas y sin conexión.
- Comprender qué tipo de información puede ponerlos ante el riesgo de robo de identidad y otras estafas.
- Reflexionar sobre las características que hacen a un ciudadano honrado.
- Diseñar resoluciones para dilemas digitales.

## Preparación

☐ Mira el video **Ciudadanía Digital - Video para el docente.**

☐ Imprime una buena selección de las plantillas de superhéroes hombre y mujer **Plantillas De Superhéroe de Cubecraft - Manipuladores** para toda la clase.

☐ Imprime una **Ciudadanía Digital - Evaluación** para cada estudiante.

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

### Para los docentes

- **Ciudadanía Digital** - Video para el docente
- **Ciudadanía Digital** - Guía de respuestas de la evaluación [Hacer una copia ▾](#)
- **Common Sense Education** - Sitio web

### Para los estudiantes

- **Plantillas De Superhéroe de Cubecraft** - Manipuladores
- **Ciudadanía Digital** - Evaluación [Hacer una copia ▾](#)

- **Diario Think Spot** - Diario de reflexiones

Hacer una copia ▼

## Vocabulario

- **Ciudadano digital** - alguien que actúa de forma segura, responsable y respetuosa en Internet.

# Guía docente

## Actividad previa (15 mín)

### Vocabulario

Esta lección tiene una frase nueva importante:

**Cuidadano Digital**- Díganlo conmigo: Ciu—da-da-no Di-gi-tal.

Alguien que actúa online con seguridad, responsablemente y respetuosamente.

### Online Personal vs. Privado

[Lesson Tip][0]

Si tiene acceso a un computador, vaya a un sitio que requiera este tipo de información, como Gmail o Facebook.

- Pregunte “¿Qué tipos de información crees que está bien compartir públicamente online, o en un perfil que otros podrán ver?”
- ¿Cuáles son algunos ejemplos de sitios webs donde debes registrarte para poder participar?
- Escriba los nombres de los sitios web en la pizarra.
- ¿Qué información se solicita y por qué crees que la piden?
- La información puede ser solicitada para ayudar a distinguir a una persona de otra.
- El sitio web puede mantener un registro de quién lo usa.
- Explique que es importante saber que compartir algún tipo de información como usuario/a, puede poner tu privacidad y la de tu familia en riesgo.
- Señale que no tienen que completar los campos en los sitios webs si no son requeridos.
- Los campos requeridos están usualmente marcados con un asterisco (\*) o destacados en rojo.
- Los/as estudiantes de escuelas básicas nunca deberían registrarse en sitios que requieran información privada, sin la aprobación y guía de uno de los padres o tutor.
- Aquí hay un ejemplo de información Pública vs. Privada:

| I SEGURO – Información Personal                    | I INSEGURO – Información Privada |
|--|----------------------------------|
| Información   \   -----                            | ----- \   -----                  |
| ----- \   Tu comida favorita                       |                                  |
| Tu opinión (aunque debe ser dada con respeto)      |                                  |
| Nombre de pila (con permiso)   Nombre de tu mamá   |                                  |
| Tu fecha de nacimiento                             |                                  |
| Información de la tarjeta de crédito de tus padres |                                  |
| Tu número de teléfono  .                           |                                  |

## Actividad Principal (30 - 40 min)



### Plantillas De Superhéroe de Cubecraft - Manipuladores

- El Hombre Araña dice: “Un gran poder conlleva una gran responsabilidad”. Esto también es así cuando trabajamos o jugamos en Internet.
- Lo que leemos, vemos y escuchamos en línea puede generar todo tipo de sentimientos en las personas (como felicidad, dolor, emoción, enojo, curiosidad).
  - Lo que hacemos o decimos en línea puede tener un impacto.
- Internet nos permite aprender acerca de todo, hablar con otras personas en cualquier momento (no importa en qué parte del mundo estén) y compartir nuestros conocimientos y proyectos creativos con los demás.



- Esto también significa que los comentarios negativos pueden llegar muy rápido a amigos de todas las edades.
- CREA un cuadro de tres columnas y escribe los términos “Segura”, “Responsable” y “Respetuosa” arriba de cada columna. Invita a los estudiantes a decir palabras o frases en voz alta que describan cómo las personas pueden actuar de forma segura, responsable y respetuosa en línea, y luego escríbelas en la columna apropiada.

| Segura | Responsable | Respetuosa |
|--------|-------------|------------|
|        |             |            |
|        |             |            |
|        |             |            |
|        |             |            |

Ahora, ¡vamos a asegurarnos de entender realmente cómo ser un Súper Ciudadano Digital!

#### Instrucciones:

- Haz que cada estudiante tome una pequeña selección de hojas de papel artístico y anímalos a combinar las piezas para crear su propio superhéroe.
- Permite que los estudiantes tengan tiempo suficiente para cortar, pegar y colorear.
- Avisa a los estudiantes cuando les queden 5 minutos para terminar.
- Forma grupos de 2 a 4 estudiantes y diles que usen sus superhéroes y los recursos que sobraron para representar una escena en la que un superhéroe observa un acto de mala ciudadanía digital. Luego, haz que el superhéroe solucione el problema ...¡y salve la situación!
- Recorre la sala y haz que cada estudiante explique su escena al resto de la clase.

#### 💡 Lesson Tip

Para ver módulos más detallados, encuentra los agregados a este plan de estudios en Alcance y secuencia de la página **Common Sense Education - Sitio web**.

## Actividad de Cierre (15 mín)

### Conversación Rápida: ¿Qué aprendimos?

- ¿Cuál es una forma correcta de actuar responsablemente online?
- ¿Qué tipos de información personal puedes compartir de ti mismo sin mostrar tu identidad?
- ¿Qué tipos de súper poderes o cualidades, tienen en común sus superhéroes digitales?
- ¿Qué significa el lema del Hombre Araña “Con gran poder, viene una gran responsabilidad” para ti, como alguien que usa internet?

#### 💡 Lesson Tip

Las preguntas de la Charla Exprés tienen como objetivo incentivar el pensamiento a gran escala, pensando en cómo la lección se relaciona con el gran mundo de los/as estudiantes y su incluso más grande futuro. Utilice el conocimiento de su aula para decidir si quiere discutirlos como clase, en grupos pequeños o compañeros/as de banco.

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil y cómo se sienten, puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

#### Sugerencias para el diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- ¿Qué es un Ciudadano Digital?
- ¿Qué necesitas hacer para ser un Ciudadano Digital?

## Evaluación (5 min)

## Ciudadanía Digital - Evaluación

- Distribuya la hoja de la evaluación y permita que los/as estudiantes completen la actividad en forma independiente luego que las instrucciones queden claras.
- Esto debería serles familiar debido a las actividades en lecciones previas.

## Aprendizaje Extendido

Usa estas actividades para mejorar el aprendizaje del alumno. Pueden ser usadas como actividades al aire libre u otros enriquecimientos.

### Common Sense Education

- Visita **Common Sense Education - Sitio web** para aprender más acerca de cómo mantener a sus alumnos seguros en esta era digital.

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

► **NI** - Redes y la Internet



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



- **Evitar a las personas agresivas** - Ficha  
[Hacer una copia ▼](#)
- **Evitar a las personas agresivas** -  
Evaluación [Hacer una copia ▼](#)
- **Poster de Seguridad Online** - Guía para el  
estudiante [Hacer una copia ▼](#)
- **Diario Think Spot** - Diario de reflexiones  
[Hacer una copia ▼](#)

## Vocabulario

- **Acosar en línea** - hacer algo en Internet, generalmente una y otra vez, para que otra persona se sienta enojada, triste o asustada.
- **En línea** - conectado a Internet.

# Guía docente

## Actividad previa (5 min)

### Introducción

Incentivar:

Incentive a los/as estudiantes a compartir lo que saben de ciberbullying.

Pregunte

- ¿Qué tipos de cosas cuentan como bullying?
- Los/as estudiantes deberían comprender que el bullying es un comportamiento que es deliberadamente malo o atemorizante para alguien más. Por ejemplo, burlarse de cómo alguien luce, contar mentiras acerca de ellos/as o hacerles algo malo.
- ¿Cómo hace sentir a otros/as el bullying?
- Heridos/as, enojados/as, molestos/as, asustados/as.
- ¿Qué es lo mejor que se puede hacer cuando te sientes que te hacen bullying, o cuándo ves a alguien más recibiendo bullying?
- Los/as estudiantes deberían saber siempre contarle a un adulto de confianza cuando experimenten o sean testigos de bullying.

## Actividad Principal (35 min)



### ¿Qué es ciberbullying?

Defina:

- *Online*: Conectado/a a internet.
- *Ciberbullying*: Hacer algo en internet, usualmente repetidamente, para hacer sentir a otra persona enojada, triste o asustada.

Discutir:

Algunos/as niños/as no usan mucho el internet, ya sea por las reglas de la familia o porque no les gusta mucho. Otros/as niños/as van online para hacer diferentes cosas.

Pregunte:

- ¿Qué hacen online, o qué crees que les gustaría hacer?
- Los/as estudiantes podrían mencionar actividades como enviar mensajes a amigos/as y jugar juegos.

Comparta:

La mayoría del tiempo cuando los/as estudiantes van online es para hacer cosas divertidas o interesantes. Pero a veces las personas pueden tratarse mal mutuamente online y esto se llama ciberbullying.

Pregunte

- ¿Viste alguna vez a alguien hacer sentir mal a otro/a online?
- Las respuestas pueden variar: Recuerde a los/as estudiantes contar lo que pasó, pero sin usar nombres reales.

Explique

Diga a los/as estudiantes que aprenderán más acerca de cómo ocurre el ciberbullying y qué hacer cuando les suceda a ellos o a alguien que conozcan.

### ¿Qué sabes acerca del Ciberbullying?



### Discutir:

Lea estos dos escenarios y discútalos brevemente con la clase.

- Kyle sigue recibiendo mensajes de alguien que dice cosas malas acerca de él. La persona que está enviando los mensajes no usa un nombre real, pero Kyle puede decir que los mensajes vienen de alguien que también se burla de él en la escuela en la clase de gimnasia.
- Sasha es nueva en la escuela, y está haciendo bastantes amigos. Entonces Sasha descubre que otra niña envió a todos/as un email que tenía una foto de una vaca con el nombre de Sasha.

Luego, distribuya la hoja de trabajo [csf-screen-out-mean][4] - Identificar al/a Malo/a de la página 6. Lea en voz alta la historia del principio y pida a los /as estudiantes trabajar en pares o grupos para terminar la hoja de trabajo.

Pida a los/as estudiantes discutir la historia de Jada. Diga a la clase que hay pasos específicos para manejar el ciberbullying.

- Jada debería DEJAR de usar el computador.
- Jada debería CONTARLE a un/a adulto/a de confianza lo que sucedió.
- Jada no debería volver al online o retornar al sitio del pony hasta que un adulto diga que está bien.
- Si Jada y Michael son buenos amigos, Jada podría querer decirle a Michael cómo sus acciones la hicieron sentir después de que ella recibió ayuda de un adulto.
- Si Michael continúa haciéndole ciberbullying, ella debería jugar con otros/as niños/as que no le hacen ciberbullying a otros/as.

En general, hay cuatro pasos que los/as estudiantes deberían tomar si ellos/as o alguien que ellos/as conocen están viviendo ciberbullying.

1. Dejar de usar el computador hasta que sea seguro.
2. Contarle a un adulto de confianza.
3. Ir online sólo cuando un adulto de confianza diga que está bien.
4. Jugar online sólo con niños/as que conoces y que son amables.

## Actividad de Cierre (15 mín)

Charla exprés: ¿qué aprendimos?

### Pregunte:

- ¿Qué es el ciberbullying? ¿Cómo hace sentir a la gente?
- Los/as estudiantes deberían reconocer que el ciberbullying es cualquier tipo de comportamiento online que hace sentir a la gente triste, asustada, enojada o molesta.
- ¿Qué cuatro cosas puedes hacer para detener el ciberbullying? S. Dejar de usar el computador hasta que sea seguro. T. Contarle a un adulto de confianza O. Ir online sólo cuando un/a adulto/a diga que está bien. P. Jugar online solo con niños/as que son amables.
- ¿Cuál es la cosa más importante que hay que hacer si alguien comienza a hacerte ciberbullying?
- Contarle a un/a adulto/a de confianza es la respuesta más importante cada vez que alguien los/as hace sentir triste, asustados/as, enojados/as online.

Escribir en el Diario

### Discussion Goal

Las preguntas para motivar la discusión incluyen:

- ¿Qué crees que pasó con el juego de Jada?
- ¿Cómo crees que Jada, Kyle, o Sasha se sintieron cuando estas cosas les pasaron?
- ¿Cómo sabes si alguien te está haciendo ciberbullying?
- ¿Por qué crees que es importante dejar de usar el computador cuando alguien comienza a hacerte ciberbullying?
- Es posible que si el/a estudiante sigue online, el ciberbullying puede continuar o volverse peor.

### Teacher Tip:

Estos escenarios se pueden leer todos juntos y discutirlos como un todo; o pueden ser leídos y discutidos de forma individual.

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil, y cómo se sienten puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

#### Sugerencias para el diario

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- Escriban los nombres de algunos/as adultos/as de confianza a los/as que puedan acudir si alguna vez sienten que les hacen bullying.
- ¿Cuáles son los cuatro pasos que deberían tomar si a ustedes o a alguien que ustedes conocen le están haciendo bullying.

## Evaluación (5 - 10 minutos)

[csf-screen-out-mean][0]

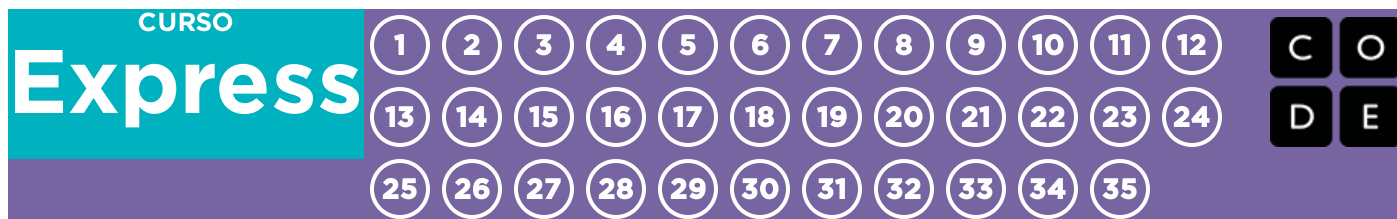
Distribuya una evaluación por estudiante. Deles tiempo para completarla y luego revise las respuestas (página 9 del link de más arriba) con la clase. Si hay tiempo, permita una discusión acerca de las preguntas.



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



# Lección 12: Eventos: El gran evento

Sin conexión | Eventos

## Descripción

Los/as estudiantes pronto aprenderán que los eventos son una gran forma de agregar flexibilidad a un algoritmo pre escrito. A veces, quiere que su programa pueda responder al/a usuario/a exactamente cuando el/a usuario/a lo quiere. Los eventos pueden hacer a su programa más interesante e interactivo.

## Propósito

Hoy, los/as estudiantes aprenderán a distinguir los eventos de las acciones. Los/as estudiantes verán acciones interrumpidas presionando un “botón” en un remoto de papel. Cuando vean este evento, la clase reaccionará con una única acción. Los eventos son ampliamente usados en programación y deberían ser fácilmente reconocibles luego de esta lección.

## Agenda

### Actividad previa (15 mín)

Vocabulario

Una serie de eventos

### Actividad Principal (15 mín)

El Gran Evento (Cursos C-F) - Controlador de Eventos

### Actividad de Cierre (10 mín)

Conversación rápida: ¿Qué aprendimos hoy?

### Evaluación (10 mín)

Eventos Sin Conexión: El Gran Evento - Evaluación

### Aprendizaje Extendido

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Repetir los comandos dados por un instructor.
- Reconocer los movimientos del docente como señales para iniciar comandos.
- Practicar la diferenciación entre acciones predefinidas y acciones basadas en eventos.

## Preparación

- ☐ Vea el **Eventos Sin Conexión: El Gran Evento - Video para el docente**.
- ☐ Imprima una **El Gran Evento (Cursos C-F) - Controlador de Eventos** y del Controlador de Eventos.
- ☐ Imprima una **Eventos Sin Conexión: El Gran Evento - Evaluación** para cada estudiante.
- ☐ Asegúrese que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones**.

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Eventos Sin Conexión: El Gran Evento - Video para el docente**
- **El Gran Evento - Guía de respuestas de la evaluación** [Hacer una copia ▼](#)

Para los estudiantes

- **Eventos Sin Conexión: El Gran Evento - Video sin conexión (descargar)**
- **El Gran Evento (Cursos C-F) - Controlador de Eventos** [Hacer una copia ▼](#)
- **Eventos Sin Conexión: El Gran Evento - Evaluación** [Hacer una copia ▼](#)



# Vocabulario

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

# Guía docente

## Actividad previa (15 mín)

### Vocabulario

Esta lección tiene una nueva e importante palabra de vocabulario:

Evento: Dígalo conmigo: E - ven - to

Un evento es una acción que causa que algo suceda.

### Una serie de eventos

- Prepare su clase para que contesten una pregunta:
  - “Voy a hacerles una pregunta. Quiero que levanten la mano si quieren que yo los elija para dar la respuesta”.
  - Haga una pregunta simple que la mayoría de sus estudiantes sean capaces de contestar, por ejemplo:
    - ¿Cuántos pulgares tengo?
    - ¿Qué es más grande, un pájaro o un caballo?
  - Señale a un/a estudiante que tenga su mano levantada y dele la oportunidad de dar su respuesta.
  - Al terminar la actividad, pregunte a los/as estudiantes cómo supo que el/a estudiante quería que lo eligiera.
    - La clase posiblemente mencione que fue porque levantaron la mano.
  - Explíquelo a todos que cuando los/as estudiantes levantan su mano, es un “evento” que causa que usted sepa que quieren ser elegidos.
- Pregunte a su clase si pueden pensar de algún otro evento que dé señales.
  - Puede que necesite recordarles que no está hablando acerca de un evento como un cumpleaños o una salida pedagógica.
  - Si tienen problemas, puede recordarles que un evento es una acción que produce que algo suceda.
    - ¿Qué hay de una alarma apagándose? ¿Qué hace que eso suceda?
    - ¿Qué hay si apretamos “comenzar” en el microondas? ¿Qué sucede?
    - ¿Qué hay si apretamos el botón “encender” en tu control remoto?
- Hoy, vamos a practicar cambiando programas introduciendo eventos.

## Actividad Principal (15 mín)

### El Gran Evento (Cursos C-F) - Controlador de Eventos

- Recuerdas cuando guiaste a Red de Angry Birds hacia el cerdo en los desafíos de Laberinto?
- En ese ejercicio, sabías por adelantado exactamente dónde querías que llegara Red, así que podías hacer un programa que llevara a Red del comienzo al final sin interrupciones.
- En la mayoría de los programas reales, no podemos hacer eso porque queremos tener opciones, dependiendo de lo que quieran los/as usuarios/as.
- Digamos que sólo quiero que mi personaje se mueva cuando mi dedo esté en la pantalla de mi teléfono. Necesitaría programar al personaje para que sólo se mueva cuando ponga mi dedo en la pantalla de mi teléfono.
- Poner mi dedo en la pantalla se convertiría en un “evento” que le dice a mi personaje que se mueva.

#### 💡 Lesson Tip

Si sus estudiantes lucen confundidos, coménteles sobre sus juegos favoritos y todas las formas en que ellos le dan a conocer a los personajes lo que se supone que deben hacer. Cuénteles lo aburrido que sería el juego si van de principio a fin sin los eventos requeridos.

En lecciones anteriores, creamos algoritmos que nos permitían controlar a un amigo u otro personaje en varios pasos a la vez. Fue divertido y útil, pero ¿qué sucede cuándo no sabes por adelantado todo lo que quieres que tu personaje haga? ¡Aquí es donde entran los eventos.

## Actividad de Cierre (10 min)

## Conversación rápida: ¿Qué aprendimos hoy?

- ¿Por qué necesitamos poder manejar los eventos en un programa?
- ¿Qué otros tipos de eventos se te ocurren?

## Evaluación (10 min)

### Eventos Sin Conexión: El Gran Evento - Evaluación

- Distribuya la hoja de la evaluación y permita que los/as estudiantes completen la actividad en forma independiente luego que las instrucciones queden claras.
- Esto debería serles familiar debido a las actividades en lecciones previas.

## Aprendizaje Extendido

Use estas actividades para ampliar el aprendizaje del estudiante/a. Pueden usarse como actividades fuera del aula u otro tipo de enriquecimientos.

### El evento de una persona es la reacción de otra

- Asigne a cada estudiante un evento del cual encargarse y una reacción apropiada para ese evento. Encadene las acciones para que así la reacción de un/a niño/a se convierta en un evento que gatilla la reacción de otro/a estudiante. Siga asignando hasta que todos tengan algo que hacer y que todos hagan reaccionar a alguien.

### Eventopalooza

- Divida la clase en grupos. Usando el controlador de eventos, asigne a cada grupo una reacción diferente al mismo botón. Haga esto para los tres botones. Luego, ¡Observe el caos!

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

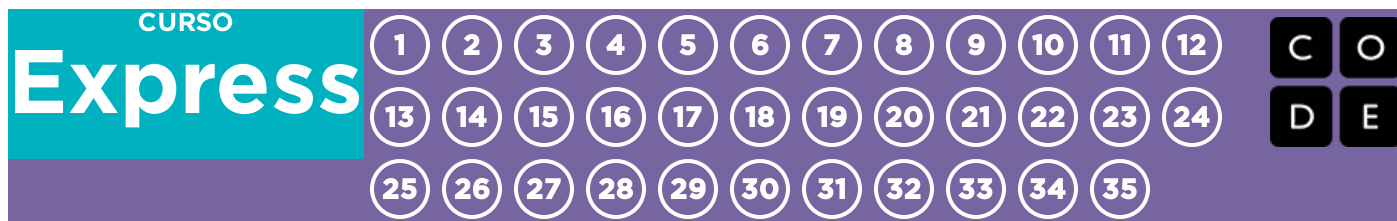
- **AP** - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



# Lección 13: Eventos en Star Wars

Star Wars | Evento

## Descripción

En esta lección, los/as estudiantes practicarán usando eventos para construir un juego que puedan compartir online. Con la participación de R2-D2 y otros personajes de Star Wars, los/as estudiantes serán guiados por los eventos, y luego se les dará espacio para crear su propio juego.

## Propósito

El Curso Fundamentos de Ciencias de la Computación no es simplemente acerca de enseñar ciencia de la computación, es acerca de hacer ciencia de la computación divertida y emocionante. En esta serie, aprenderán acerca de los eventos usando personajes populares de Star Wars. Estos desafíos borran un poco la línea entre “aprendizaje” y “diversión”. También, aprenderán a reconocer prácticas de programación regulares en juegos para que cuando jueguen en casa, puedan ver los principios de la ciencia de la computación usados.

## Agenda

### Actividad previa (15 mín)

Introducción

### Actividad Puente - Eventos (15 mín)

Actividad sin conexión con bloques de papel  
Previsualizar los desafíos en línea con la clase

### Actividad Principal (30 min)

Curso Rápido Fundamentos de Ciencias de la Computación  
- Sitio web

### Actividad de Cierre (15 mín)

Escribir en el Diario

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

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

## Preparación

- ☐ Juegue con los desafíos **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web** para encontrar cualquier potencial área problemática para su clase..
- ☐ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones**.
- ☐ Asegúrese de cada estudiante tenga un **Diario Think Spot - Diario de reflexiones**.

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso E Desafíos Online** - Sitio web
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación** - Recomendaciones de lecciones [Hacer una copia ▾](#)

Para los estudiantes

- **Bloques Blockly Sin Conexión (Cursos 2-5)** - Manipuladores
- **Diario Think Spot** - Diario de reflexiones [Hacer una copia ▾](#)

## Vocabulario

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

# Guía docente

## Actividad previa (15 mín)

### Introducción

En una discusión en la clase, pregunte a los/as estudiantes cuál es su video juego favorito (puede que tenga que recordarles sólo usar juegos que son apropiados para la sala de clases). Pregúnteles cuál es su parte favorita del juego.

La mayoría de las veces, los/as estudiantes responderán con algún tipo de evento. Cuando reconozca la respuesta de un/a estudiante que describe un evento, pregúntele que lo describa más en detalle.

Una vez que el/a estudiante ha terminado de describirlo, tómese un minuto para relacionarla con la definición de un evento.

- **Evento:** Una acción que causa que algo suceda.

Pídales que intenten relacionar algunas de sus partes favoritas de los video juegos y cómo pueden ser descritos como eventos. Hágalos compartir en parejas y discutir las diferencias entre sus eventos y los de su compañero/a.

#### Teacher Tip

Si no está muy seguro/a que la respuesta del/a estudiante describe un evento, intente desglosar la respuesta. ¿Hay una acción y una respuesta?

Por ejemplo:

- Cruzar la línea final y tener personajes en pantalla que te felicitan.
- Encontrar un gran recipiente con tesoros (u otro objeto) y ver tu inventario crecer.
- Comprar nuevos objetos en la tienda de juegos y usar esos objetos.
- Presionar los botones en el control del juego y hacer que tu personaje haga algo genial.

## Actividad Puente - Eventos (15 mín)

Esta actividad te ayudará a llevar los conceptos sin conexión de “El gran evento” al mundo en línea en el que se mueven tus estudiantes. Elige *una* de las siguientes actividades para realizar con tu clase:

### Actividad sin conexión con bloques de papel

Con el control remoto de **El Gran Evento (Cursos C-F) - Controlador de Eventos y Bloques Blockly Sin Conexión (Cursos 2-5) - Manipuladores**, reúne a tu clase para repetir la actividad de la lección previa. Pregunte a la clase: “¿Qué hacemos cuando se pulsa el botón turquesa?”. Luego, completa uno de los bloques de evento cuando y uno de los bloques de acción azules. Asegúrate de que los estudiantes entiendan que los bloques cuando deben estar encima del bloque azul y en contacto para que el programa se ejecute.

### Previsualizar los desafíos en línea con la clase

Elige una lección de la etapa en línea correspondiente de **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**; te recomendamos un desafío previo, como el segundo. Pregunte a los estudiantes qué debe suceder cuando se presiona la flecha hacia arriba en el teclado. Explica que el personaje en este juego se moverá en la dirección de la flecha.

Completa el desafío con la clase y permite que debatan rápidamente sobre qué fue y qué no fue un evento. Para cada evento, pregunte a los estudiantes cuál es la acción que corresponde.

## Actividad Principal (30 min)

### Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

Es posible que los estudiantes estén muy emocionados de crear su propio juego de Star Wars al final de este grupo de desafíos. Si hay tiempo, pídeles que planifiquen lo que quieren que haga el juego. La planificación y preparación los ayudará a reconocer mejor los principales conceptos que enseña esta lección. Anímalos a compartir y combinar los juegos de los demás al final de la lección.

#### Teacher Tip

Recuerda a los estudiantes que solo compartan sus trabajos con amigos cercanos o familiares. Para más información, muestra a la clase:

**Detente y Piensa Online - Video.**

## Actividad de Cierre (15 mín)

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil y cómo se sienten, puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el Diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- Da un ejemplo de un evento que utilizaste en tu programa hoy?
- ¿Por qué es importante no compartir información privada online? ¿Cómo sabes si la información es privada?

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

- **AP** - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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

# Lección 14: Eventos con Flappy

Evento | Flappy

## Descripción

En esta etapa especial, los/as estudiantes pueden crear su propio juego Flappy Bird utilizando manejadores de eventos para detectar clics del ratón y colisiones de objetos. Al final del nivel, los/as estudiantes podrán personalizar su juego cambiando las imágenes o las reglas.

## Propósito

Los eventos son muy comunes en los programas de computador. En esta lección, los/as estudiantes desarrollarán su comprensión de los eventos mediante la creación de un juego de Flappy Bird. Los/as estudiantes aprenderán a hacer que su personaje se mueva por la pantalla, a hacer ruidos y a reaccionar ante los obstáculos basados en eventos iniciados por el usuario.

## Agenda

### Actividad previa (10 min)

Introduction

### Actividad Puente - Eventos (10 min)

Actividad Sin conexión Usando Bloques de Papel

Pre visualización de los desafíos en línea con la clase

### Actividad Principal (30 min)

Curso Rápido Fundamentos de Ciencias de la Computación  
- Sitio web

### Actividad de Cierre (10 - 15 min)

Escribir en el Diario

### Aprendizaje Extendido

## Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Hacer coincidir los bloques con el controlador de eventos apropiado.
- Crear un juego mediante controladores de eventos.
- Compartir un artefacto creativo con otros estudiantes.

## Preparación

- ▣ Juegue con **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web** en etapa 3 para encontrar cualquier potencial área problemática para su clase.
- ▣ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones.**
- ▣ Asegúrese de cada estudiante tenga un **Diario Think Spot - Diario de reflexiones.**

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones** [Hacer una copia ▾](#)

Para los estudiantes

- **Diario Think Spot - Diario de reflexiones** [Hacer una copia ▾](#)

## Vocabulario



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

# Guía docente

## Actividad previa (10 min)

### Introduction

- Repase la actividad de "El gran evento" con los estudiantes: - ¿Para qué "programamos" el botón para hacer clic en los eventos?
- Ahora vamos a agregar eventos a nuestra codificación. Específicamente, vamos a crear un evento para hacer clic con el mouse y otro para cuando el pájaro golpee un objeto como el suelo o un obstáculo. ¿Cuándo has visto a un personaje tocar otro objeto como un evento en los juegos?

#### 💡 Tips para la Lección

Los estudiantes tendrán la oportunidad de compartir su producto final con un enlace. Esta es una gran oportunidad para mostrar a su comunidad escolar las grandes cosas que hacen sus estudiantes. ¡Monitoree y recopile todos los enlaces a los proyectos y manténgalos en el sitio web de su clase para que todos los vean!

## Actividad Puente - Eventos (10 min)

Esta actividad le ayudará con los conceptos sin conexión del "Gran Evento" hacia el mundo online al que sus estudiantes se están moviendo. Elija uno de los siguientes para realizar con su clase:

### Actividad Sin conexión Usando Bloques de Papel

Usando el control de **El Gran Evento (Cursos C-F) - Controlador de Eventos y Bloques Blockly Sin Conexión (Cursos 2-5) - Manipuladores** reúna a su clase para repetir la actividad de la lección previa. Pregunte a la clase "¿Qué hacemos cuando se pulsa el botón verde azulado?". Luego rellene uno de los bloques de "evento cuando", y uno de los bloques de acción azules. Asegúrese de que los/as estudiantes entiendan que los bloques "cuando" necesitan estar encima del bloque azul y que se necesitan tocar para que el programa se ejecute.

### Pre visualización de los desafíos en línea con la clase

Elija una lección de la etapa en línea, te recomendamos el desafío 2. Pregunte a los/as estudiantes qué debe suceder cuando el Flappy Bird choca con algo como el suelo o un obstáculo. Explique que Flappy en este juego se moverá hacia adelante con un clic del ratón y el juego terminará si Flappy se topa con algo.

Complete el desafío con la clase y dé tiempo para una discusión rápida sobre lo que fue y lo que no fue un evento. Para cada evento, pregunte a los/as estudiantes cuál es la acción que corresponde.

## Actividad Principal (30 min)

### Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

En la etapa final de esta lección, los estudiantes pueden ajustar su juego para que sea único; anímalos a ver qué tan lejos pueden llegar dentro de las restricciones provistas. Si la clase no usa **Programación en Parejas - Video para el estudiante**, entonces díles a los estudiantes que miren los juegos de los demás. De lo contrario, haz que los estudiantes debatan y prueben diferentes formas de configurar su juego con su compañero.

#### 💡 Teacher Tip

Recuerda a los estudiantes que solo compartan sus trabajos con amigos cercanos o familiares. Para más información, muestra **Detente y Piensa Online - Video** a la clase:

## Actividad de Cierre (10 - 15 min)

## Escribir en el Diario

Hacer que los alumnos escriban acerca de lo que han aprendido, por qué es útil, y cómo se sienten puede ayudar a solidificar cualquier conocimiento obtenido hoy y construir una hoja de revisión que pueden revisar en el futuro.

Sugerencias para el Diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- ¿Qué hiciste para hacer tu juego único?
- Dibuja un juego que quieres hacer en el futuro.

## Aprendizaje Extendido

Utilice estas actividades para mejorar el aprendizaje de los/las estudiantes. Pueden ser utilizados como actividades fuera de la clase o en otra oportunidad.

### Mira Bajo la Capa

Cuando compartes un link para tu juego, también estás compartiendo el código detrás de él. Esta es una excelente forma de que los/as estudiantes aprendan los/as unos/as de los/as otros/as.

- Publique los links de los juegos online terminados o en la pizarra.
  - ¡Haga usted mismo/a un juego para compartir también!
- Cuando los/as estudiantes suben un link, deben hacer clic en el botón “Cómo funciona” para ver el código detrás del juego.
- Discutir en grupos las diferentes formas utilizadas por sus compañeros/as para codificar sus juegos.
  - ¿Qué te sorprendió?
  - ¿Qué te gustaría probar?
- Elija el juego de alguno/a de sus estudiantes y trabaje sobre él. (No se preocupe; el juego original estará guardado)

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

- **AP** - Algoritmos y programación



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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

# Lección 15: Eventos en Bounce

Evento | Rebotar

## Descripción

En esta actividad online, los/as estudiantes aprenderán cuáles son los eventos y cómo los computadores los utilizan en programas como videojuegos. Los/as estudiantes trabajarán a través de desafíos y harán que el programa reaccione a eventos (como presionar los botones de flecha.) Al final del desafío, los/as estudiantes tendrán la oportunidad de personalizar su juego con sonidos y diferentes velocidades.

## Propósito

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

En esta lección, los/as estudiantes desarrollarán su comprensión de los eventos haciendo un juego basado en el deporte. Los/as estudiantes aprenderán a hacer que su paddle se mueva de acuerdo con las teclas de flecha, y a hacer ruidos cuando los objetos chocan. Al final, ellos tendrán que personalizar su juego para hacerlo más única!

## Agenda

**Actividad previa (10 min)**

**Actividad Principal (30 min)**

Curso Rápido Fundamentos de Ciencias de la Computación  
- Sitio web

**Actividad de Cierre (10 min)**

Escribir en el Diario

**Aprendizaje Extendido**

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

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

## Preparación

- ☐ Juegue **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web** para encontrar áreas de potenciales problemas para su clase.
- ☐ Asegúrese que cada estudiante tenga un **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones.**
- ☐ Revise **Diario Think Spot - Diario de reflexiones.**

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones** [Hacer una copia](#)

Para los estudiantes

- **Diario Think Spot - Diario de reflexiones** [Hacer una copia](#)

## Vocabulario

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

# Guía docente

## Actividad previa (10 min)

Pida a los/as estudiantes que se sienten cerca de usted. ¡Ahora pídeles que se pongan de pie!

Diga a los/as estudiantes que lo que acaba de hacer fue declarar un evento y una acción. Cuando usted les dice que se sienten, eso es un evento. La acción que responde a este evento es que la clase se sienta. Esto es lo mismo cuando les pide que se pongan de pie. Los eventos y las acciones son muy fácilmente identificables en nuestras vidas.

Algunos de los eventos y de las acciones incluyen:

- Tener hambre y comer.
- Golpearle el dedo y gritar “¡Auch!”
- ¡Pasar la pelota de basquetbol por el aro y marcar un punto para tu equipo!

Pídale a la clase que piensen un par de eventos más. ¡Cuénteles que van a hacer un juego donde el programa tendrá acciones asociadas a los eventos que codifiquen!

## Actividad Principal (30 min)

### Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

Al final del grupo de desafíos, los estudiantes tendrán la oportunidad de hacer que su juego sea único.

Pídeles que prueben nuevas formas de hacer más difícil su juego; por ejemplo, que intenten jugar con varias pelotas a la vez o que se lancen más pelotas cada vez que la pelota rebota en el muro.

Esto es lo que **hizo Justin Trudeau, el primer ministro de Canadá, cuando completó una Hora del Código en 2016.**

#### 💡 Teacher Tip

Recuerda a los estudiantes que solo deben compartir su trabajo con amigos cercanos y familiares. Para más información, muestra **Detente y Piensa Online - Video** a tu clase.

## Actividad de Cierre (10 min)

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil y cómo se sienten, puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el Diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- ¿Qué hiciste para que tu juego fuera súper genial?
- ¿Qué tipo de juego quieres codificar en el futuro?

## Aprendizaje Extendido

Llévame al Juego de Pelota.

Lleve a los/as estudiantes al exterior para jugar un tipo de juego de pelota. Lleve registro de los eventos y de las acciones. Por ejemplo, no hacer “dribbling” en basquetbol resulta en una infracción y el otro equipo se queda con la pelota. En fútbol, patear la pelota fuera de los límites resulta en que el otro equipo pateee la pelota hacia la cancha. ¡Llevar la pelota al arco resulta en un punto! Invente más eventos si sus estudiantes lo disfrutan. Pida a todos/as las estudiantes que griten “Yipii” cuando el capitán de un equipo marque un punto. ¡Haga que todo caigan al suelo y rueden si un /a estudiante hace dos goles seguidos!

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

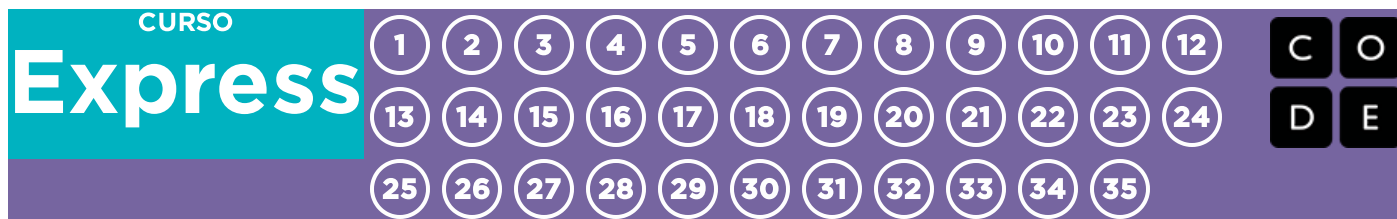
► **AP** - Algoritmos y programación



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



# Lección 16: Condicionales: condicionales con naipes

Condicionales | Sin conexión

## Descripción

Esta lección demuestra cómo se pueden usar los condicionales para moldear un programa a alguna información específica. No siempre tenemos toda la información que necesitamos cuando escribimos un programa. A veces querrá hacer algo diferente en una situación y en otra, incluso si no sabe cuál situación será verdadera cuando ejecute el código. Ahí es donde entran los condicionales. Los condicionales permiten al computador tomar una decisión, basada en la información que es verdadera cada vez que su código se está ejecutando.

## Propósito

Una de las mejores formas de enseñar condicionales es que los/as estudiantes ya comprenden este concepto en su vida diaria. Esta lección mezcla la ciencia de la computación con el mundo real, mediante la construcción de su habilidad de decir si una condición es verdadera o falsa. Los/as estudiantes aprenderán a usar las afirmaciones “Si” para declarar cuando cierto comando debe ser ejecutado, así como afirmaciones “Si / Si no”, para declarar cuando se debe ejecutar un comando y que debe ser ejecutado de otra manera. Puede que los/as estudiantes no reconozcan la palabra condicionales, pero la mayoría de los/as estudiantes comprenderán la idea de usar “Si” para asegurarse que algunas acciones sólo ocurran cuando se supone que deben hacerlo.

## Agenda

### Actividad previa (20 min)

Vocabulario  
Introducción

### Actividad Principal (20 min)

Programa Condicionales con Cartas de Muestra - Guía de preparación para el docente

### Actividad de Cierre (15 min)

Conversación Rápida: ¿Qué aprendimos?  
Escribir en el Diario

### Evaluación (5 min)

Si/si no: Condicionales con Cartas - Evaluación

### Aprendizaje Extendido

## Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Definir las circunstancias en que ciertas partes de un programa deben ejecutarse y cuándo no.
- Determinar si se cumple un condicional según los criterios.
- Recorrer un programa y predecir el resultado dado un conjunto de información.

## Preparación

- ☐ Vea **Si/si no: Condicionales con Cartas - Video para el docente.**
- ☐ Vea **Si/si no: Condicionales con Cartas - Video de lección en acción.**
- ☐ Junte mazos de cartas o algo similar.
- ☐ Una **Programa Condicionales con Cartas de Muestra - Guía de preparación para el docente** para mostrar a la clase.
- ☐ Imprima una copia **Si/si no: Condicionales con Cartas - Evaluación** para cada estudiante.
- ☐ Asegúrese que cada estudiante tiene un **Diario Think Spot - Diario de reflexiones.**

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Si/si no: Condicionales con Cartas - Video** para el docente
- **Si/si no: Condicionales con Cartas - Video** de lección en acción
- **Programa Condicionales con Cartas de Muestra - Guía de preparación para el docente** [Hacer una copia ▼](#)



- **Si/si no: Condicionales con Cartas** - Video de la evaluación
- **Condicionales con Cartas** - Guía de respuestas de la evaluación

Hacer una copia ▼

Para los estudiantes

- **Si/si no: Condicionales con Cartas** - Video sin conexión (**descargar**)
- **Si/si no: Condicionales con Cartas** - Evaluación
- **Diario Think Spot** - Diario de reflexiones

Hacer una copia ▼

## Vocabulario

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

# Guía docente

## Actividad previa (20 min)

### Vocabulario

Esta lección tiene una nueva palabra:

**Condicionales** – Díganlo conmigo: Con-di-cio-na-les

Afirmaciones que sólo se ejecutan bajo ciertas condiciones.

### Introducción

- Podemos comenzar esta lección de inmediato.
- Diga a la clase que si pueden estar completamente en silencio por treinta segundos, usted hará una de las siguientes acciones:
- Cantar una canción de una ópera.
- Darles cinco minutos más de recreo.
- Hacer la posición invertida.
- Comience a contar de inmediato.
- Si los/as estudiantes tienen éxito, señale que lo hicieron muy bien, así que obtendrán su recompensa.
- De lo contrario, señale que no completaron el tiempo de treinta segundos de silencio, así que no obtendrán su recompensa.
- Pregunte a la clase “¿Cuál fue la condición para la recompensa?”
- La condición fue Si estaban en silencio por 30 segundos.
- Si lo estuvieron, la condición sería verdadera, y obtendrían la recompensa.
- Si no lo estuvieron, la condición sería falsa, y la recompensa no aplicaría.
- ¿Podemos crear otro condicional?
- Si pueden adivinar mi edad, la clase les puede aplaudir.
- Si sé una respuesta, puedo levantar mi mano.
- ¿Qué ejemplos puedes crear?
- A veces, queremos tener una condición extra, en caso que la afirmación “Si” no fuera verdadera.
- Esta condición extra se llama una afirmación “Si no”.
- Cuando la condición “Si” no se cumple, podemos mirar al “Si no” para saber qué hacer.
- Ejemplo: Si dibujo un rey de este mazo de cartas, todos aplauden. Si no, todos dicen “Ooooooh”.
- Intentémoslo. (Dibuje una carta y vea si su clase reacciona apropiadamente)
- Pida a la clase analizar lo que sucedió.
- ¿Qué fue el “Si”?
- ¿Qué fue el “Si no”?
- ¿Qué condición se cumplió?
- Aunque no lo crean, tenemos incluso una opción más.
- ¿Qué pasaría “Si” ustedes aplauden cuando saco un 7 del mazo, “Si no” dicen “¡YAY!”, cuando saco un número menor que siete, o “Si” ninguna de las anteriores se cumple ustedes dicen “¡oooooh!”?
- Por esto tenemos el término “Si”, “Si / Si no” y “Si no”.
- “Si” es la primera condición.
- “Si no / Si” se usa sólo cuando “Si” no es verdad.
- “Si” se usa sólo cuando nada de lo anterior es verdad.

Ahora, hagamos un juego.

## Actividad Principal (20 min)

### Programa Condicionales con Cartas de Muestra - Guía de preparación para el docente

### Instrucciones:

- Crea unos pocos programas con tu clase, que dependan de cosas como el palo, el color o el valor de los naipes, para sumar o restar puntos. Puedes escribir el programa como un algoritmo, un pseudocódigo o un código real.

Este es un ejemplo de algoritmo:

```
Si (el NAIPE es ROJO)
  Premia a TU equipo con 1 punto

Si no
  Premia al OTRO equipo con 1 punto
```

Este es un ejemplo del mismo programa en pseudocódigo:

```
Si (naipe.color == ROJO){
  puntos.tuyos = puntos.tuyos + 1;
}

Si no {
  puntos.otros = puntos.otros + 1;
}
```

- Decide cómo dividirás a tu clase en equipos.
- Cada equipo debería tener una pila de naipes (al menos tantos naipes como miembros del equipo) a mano.
- Pon uno de tus “Programas” en la pizarra para que todos lo vean.
- Haz que los equipos tomen naipes por turnos y sigan el programa para ver cuántos puntos obtienen en cada ronda.
- Juega varias veces con diferentes programas para ayudar a los estudiantes a entender los condicionales.

Una vez que la clase ha practicado, puedes animar a los estudiantes a anidar los condicionales uno dentro de otro:

```
Si (el NAIPE es ROJO)
  Premia a TU equipo con 1 punto

Si no
  Si (el NAIPE es mayor que 9)
    Premia al OTRO equipo con 1 punto
  Si no
    Premia a TU equipo con el mismo número de puntos del naipe
```

Este es el mismo programa en pseudocódigo:

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

## Actividad de Cierre (15 mín)

### Conversación Rápida: ¿Qué aprendimos?

- Si iban a codificar esto en Blockly, ¿qué necesitarían agregar a sus condicionales para que el código se ejecute más de una vez? (loop)
- ¿Qué otras cosas hacen durante el día, bajo ciertas condiciones?
- Si se supone que hagan algo cuando el valor de la carta es más que 5, y sacas un 5, ¿cumples esa condición?
- Note que las condiciones pueden ser “Verdaderas” o “Falsas”. No hay evaluación de una condición que evalúe al “Plátano”.
- Cuando necesiten cumplir varias combinaciones de condiciones, podemos usar algo llamado “condicionales anidados”.
  - ¿Qué crees que significa eso?

#### 💡 Lesson Tip

Las preguntas de la Charla Exprés tienen como objetivo incentivar el pensamiento a gran escala pensando en cómo la lección se relaciona con el gran mundo de los/as estudiantes y su incluso, más grande futuro. Utilice el conocimiento de su aula para decidir si quiere discutirlos como clase, en grupos pequeños o compañeros/as de banco.

- ¿Puedes dar un ejemplo de dónde vimos eso durante el juego?
- ¿Qué parte del juego te gustó más?

## Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil y de cómo se sienten, puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- ¿Qué es un condicional? ¿Cómo usaste un condicional hoy?
- ¿Cuáles son algunos condicionales que usaste hoy? ¿Puedes crear algunos más que podrías usar con un mazo de cartas?

## Evaluación (5 min)

### Si/si no: Condicionales con Cartas - Evaluación

Distribuya la hoja de la evaluación y permita que los/as estudiantes completen la actividad en forma independiente luego que las instrucciones queden claras. Esto debería serles familiar debido a las actividades en lecciones previas. Aquí hay un **Si/si no: Condicionales con Cartas - Video de la evaluación** para ver un video como guía.

## Aprendizaje Extendido

Use estas actividades para mejorar el aprendizaje de sus estudiantes. Pueden usarse como actividades fuera de aula u otros enriquecimientos.

### Etiqueta Verdadero/Falso

- Ponga a los/as estudiantes en fila para jugar Luz Roja/Luz Verde.
- Seleccione a una persona para pararse delante como el que dirige.
- El que dirige escoge una condición y le pregunta a todos quien cumple la condición para que dé un paso al frente.
- Si tienen un cinturón rojo, un paso al frente.
- Si estás usando sandalias, un paso al frente.
- Intente cambiar diciendo cosas como “si te gustan las manzanas, un paso al frente”.

### Anidar

- Separe a los/as estudiantes en parejas o pequeños grupos.
- Pídales que escriban afirmaciones “Si” en tiras de papel para jugar cartas, como:
  - la pinta es trébol.
  - el color es rojo.
- Pida a los/as estudiantes crear tiras de papel para los resultados.
- Agregar un punto.
- Quitar un punto.
- Una vez que eso está hecho, pida a los/as estudiantes que escojan tres de cada tipo de tira y tres cartas, poniendo atención al orden seleccionado.
- Usando estos papeles, pida a los/as estudiantes escribir tres programas diferentes usando sólo las tiras de papel seleccionadas, en cualquier orden.
- Anime a los/as estudiantes a poner algunas afirmaciones “Si” dentro de otras afirmaciones “Si”.
- Ahora, los/as estudiantes deberían ejecutar los tres programas usando las cartas que escogieron, en el mismo orden para cada programa.
- ¿Entregó la misma respuesta, alguno de los dos programas?
- ¿Alguno respondió algo diferente?

# Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

► **AP** - Algoritmos y programación



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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

# Lección 17: Loops Mientras en la Granjera

Bucles mientras | Bucles | Granjera

## Descripción

Para el momento en que los/as estudiantes lleguen a esta lección, ya deberían haber practicado lo suficiente usando loops repetir, así que ahora es tiempo de mezclar las cosas. Los loops mientras son loops que continúan repitiendo comandos mientras se cumpla una condición. Los loops mientras se usan cuando el/la programador/a no sabe el número exacto de veces que es necesario repetir los comandos, pero sí sabe qué condición necesita ser cierta para que el loop continúe repitiendo. Por ejemplo, los/as estudiantes trabajarán llenando hoyos y sacando tierra en Granjera. No sabrán el tamaño de los hoyos ni la altura de las montañas de tierra, pero sabrán que necesitan seguir llenando los hoyos y sacando tierra mientras el suelo no esté plano.

## Propósito

A medida que sus estudiantes sigan profundizando sus conocimientos acerca de los loops, encontrarán problemas en los que un comando debe repetirse, pero es desconocido cuántas veces necesita repetirse. Aquí es donde vienen loops mientras. En la lección de hoy, los estudiantes desarrollarán la comprensión de los loops basados en las condiciones y también ampliarán su conocimiento de loops en general.

## Agenda

### Actividad previa (10 min)

Introducción

### Actividad Puente - Condicionales (15 min)

Actividad Sin Conexión Usando Bloques de Papel  
Visión previa de los desafíos en línea

### Actividad Principal (30 min)

Curso Rápido Fundamentos de Ciencias de la Computación  
- Sitio web

### Actividad de Cierre (15 min)

Escribir en un diario

### Aprendizaje Extendido

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Distinguir entre bucles que se repiten un número fijo de veces y bucles que se repiten siempre que una condición sea verdadera.
- Usar un bucle mientras para crear programas que puedan resolver problemas con valores desconocidos.

## Preparación

- ▢ Juegue **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web** para encontrar áreas de potenciales problemas para su clase.
- ▢ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones.**
- ▢ Asegúrese que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones.**

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones** [Hacer una copia](#)

Para los estudiantes

- **Bloques Blockly Sin Conexión (Cursos 2-5)**  
- Manipuladores
- **Diario Think Spot** - Diario de reflexiones  
[Hacer una copia](#)

# Vocabulario

- **Condición** - una sentencia que un programa verifica si es verdadera o falsa. Si es verdadera, se realiza una acción. De lo contrario, se ignora.
- **Bucle** - la acción de hacer algo una y otra vez.
- **Repetir** - hacer algo de nuevo.
- **Bucles mientras** - un bucle que se sigue repitiendo mientras una condición es verdadera.

# Guía docente

## Actividad previa (10 min)

### Introducción

Usa “mientras” en una oración en frente de la clase. Pregúntales qué significa la palabra “mientras”. Si tú dijeras “mientras exista un hoyo, llénalo con tierra”, ¿qué harían?, ¿por cuánto tiempo lo harían?

Cuando usas una palabra como “mientras”, estás confiando en una condición que le diga a la computadora por cuánto tiempo debería ejecutarse el bucle. Una condición es una afirmación que se prueba y se descubre si es verdadera o falsa. En el caso anterior, la condición es ‘si hay un hoyo’. Solo es posible que exista un hoyo o que no exista un hoyo, de modo que la afirmación puede ser solo verdadera o falsa.

Cuéntales a los estudiantes que aprenderán sobre un nuevo tipo de bucles. Previamente, los estudiantes solo usaron bucles para repetir un comando un cierto número de veces. Aquí, no siempre sabrán cuántas veces repetir el comando; sin embargo, sabrán cuándo detenerse o cuándo continuar. Los bucles ‘mientras’ le permiten al quien programa repetir un comando siempre que una condición sea verdadera.

Si hay tiempo, haga que los estudiantes discutan otras veces en las que el uso de un “bucle mientras” sería útil. Por ejemplo:

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

## Actividad Puente - Condicionales (15 mín)

Esta actividad ayudará a traer los conceptos sin conexión de los "Condicionales con Cartas" al mundo en línea al que los alumnos están entrando. Elige uno de los siguientes para hacer con su clase:

### Actividad Sin Conexión Usando Bloques de Papel

Imprima y corte 2 -3 bloques 'si/más' y de acción en blanco de **Bloques Blockly Sin Conexión (Cursos 2-5) - Manipuladores** y saca un mazo de cartas. Pide a la clase que propongan un par de condicionales para usar con el mazo de cartas como lo hicieron en "Condicionales con Cartas". Cuando los condicionales hayan sido elegidos como clase, completa la parte en blanco del bloque 'si' con distintos valores de las cartas que los niños propusieron. Algunos ejemplos incluyen "Rey de Corazones", "Números Pares" o "Diamantes". Completa los bloques de acción con las acciones que los alumnos propusieron. Asegúrate que los alumnos conozcan los bloques de acción que se necesitan directamente bajo el bloque 'si' o 'más'. Más abajo se muestra un ejemplo.



Ahora mezcla el mazo de cartas y juega nuevamente "Condicionales con Cartas". Gira el mazo carta por carta, reaccionando a las cartas si se ha hecho un condicional para ella.

### Visión previa de los desafíos en línea

Saca un desafío del Curso D, recomendamos el desafío 9.

- Pregunta al curso lo que la abeja debería hacer cuando llegue a la nube.
  - La abeja debería usar un condicional para revisar si hay una flor o un panal.



- Usa el bloque 'si en la flor/más'. Pregunta a la clase que debería hacer la abeja si hay una flor. si no hay una flor, habrá un panal. ¿Qué debería hacer la abeja entonces?
  - La abeja debería 'recoger néctar' si hay una flor y 'hacer miel' si hay un panal.

Completa el resto del código y presione 'ejecutar'. Discute con la clase sobre por qué esto funcionó.

## Actividad Principal (30 min)

### Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

Los “bucles mientras” no siempre son un concepto difícil de entender para los estudiantes, pero si crees que tu clase podría tener problemas con estos desafíos, recomendamos **Programación en Parejas - Video para el estudiante**. Esto les permitirá a los estudiantes compartir las ideas de todos antes de implementar el código. La programación en parejas sirve para incrementar la confianza y la comprensión de los temas como los “bucles mientras”.

## Actividad de Cierre (15 mín)

### Escribir en un diario

Hacer que los estudiantes escriban acerca de lo que aprendieron, por qué es útil, y cómo se sienten. Puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- ¿Cuál es la diferencia entre un bucle mientras y un bucle repetir normal?
- Brinda un ejemplo de un desafío donde usarías un bucle mientras, pero no un bucle repetir. ¿Puedes dar un ejemplo de un desafío donde usarías un bucle repetir, pero no un bucle mientras?

## Aprendizaje Extendido

Simón dice “Mientras”

Sal al patio y pon a los estudiantes en fila frente de ti. Asegúrate de que cada estudiante pueda verte. Declara un par de “bucles mientras”, como: • Mientras mi mano derecha esté arriba, pueden caminar hacia mí. • Mientras cubro mis ojos, pueden saltar hacia mí. • Mientras mi cabeza este girada hacia la derecha, tienen que caminar hacia atrás.

El primer estudiante que llegue a ti, gana. Si hay tiempo, deja que otros estudiantes hagan de “Simón” frente a la clase.

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

- AP - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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

# Lección 18: Condicionales & Loops en Laberinto

Condicional | Bucle | Laberinto | Angry Bird | Zombi

## Descripción

En esta lección, los/as estudiantes unirán dos conceptos clave: loops y condicionales. Este conjunto de desafíos disminuye las brechas en la comprensión de qué ocurre al trabajar en desafíos que usan múltiples tipos de bloques. Al unir dos ideas, los/as estudiantes crearán un código más complejo que muestra impresionante creatividad y pensamiento crítico!

## Propósito

Este conjunto de desafíos servirá para solidificar y construir sobre el conocimiento de los condicionales y loops. Al combinar estos dos conceptos, los/as estudiantes podrán explorar el potencial para crear programas complejos e innovadores en un ambiente nuevo y emocionante.

## Agenda

### Actividad previa (10 min)

Introducción

### Actividad Principal (30 min)

Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

### Actividad de Cierre (15 min)

Escribir en el Diario

### Aprendizaje Extendido

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Desarrollar programas con la comprensión de varias estrategias para implementar condicionales.
- Convertir sentencias condicionales de lenguaje hablado y bucles en un programa.

## Preparación

- ▢ Juegue **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web** para encontrar áreas de potenciales problemas para su clase.
- ▢ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones.**
- ▢ Asegúrese que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones.**

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones** [Hacer una copia ▾](#)

Para los estudiantes

- **Diario Think Spot - Diario de reflexiones** [Hacer una copia ▾](#)

## Vocabulario

- **Condición** - una sentencia que un programa verifica si es verdadera o falsa. Si es verdadera, se realiza una acción. De lo contrario, se ignora.
- **Condicionales** - sentencias que solo se ejecutan bajo ciertas condiciones.
- **Bucle** - la acción de hacer algo una y otra vez.
- **Repetir** - hacer algo de nuevo.
- **Bucles mientras** - un bucle que se sigue repitiendo mientras una condición es verdadera.

# Guía docente

## Actividad previa (10 min)

### Introducción

Los/as estudiantes unirán los nuevos “Loops mientras” y las afirmaciones “si/si no”, así que esta instrucción repasará lo que hacen estos bloques en un sentido general.

Loops Mientras

Pregunte a la clase si recuerdan lo que significa “mientras” en codificación. Repase la definición de un “loop mientras”.

- *Loops Mientras*: Un loop que continúa repitiendo mientras una condición sea verdadera.

Además de los loops mientras, los/as estudiantes practicarán con los “loops hasta” en este grupo de desafíos. Explíqueles que los “loops mientras” continúan repitiendo el código mientras una condición sea verdadera, mientras los loops hasta continúan hasta que una condición es verdad.

Por ejemplo, con un loop mientras, el zombi continuará caminando por el camino mientras haya un camino por delante. Con un loop hasta, el zombi continuará caminando hasta que alcance la flor al final del camino. Eso se mostrará más en detalle dentro de los desafíos.

## Actividad Principal (30 min)

### Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

Relacionar los conceptos no es fácil, pero este grupo de lecciones está pensado para unir las sentencias `si / si no` con los bucles `mientras`, y que los estudiantes descubran las infinitas posibilidades de programar que tienen cuando se usan juntos. Si los estudiantes tienen problemas para comprender las similitudes o diferencias entre las sentencias `si / si no`, los bucles `mientras` o los bucles `hasta`, pídeles que vuelvan a practicar los desafíos previos que solo usan uno de los tres.

## Actividad de Cierre (15 mín)

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil, y cómo se sienten. Puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- ¿Cuál es la diferencia entre un loop "hasta" y un loop "mientras"?
- ¿Qué te gusta usar más: afirmaciones "si/si no", loops "hasta" o loops "mientras"? ¿Por qué?

## Aprendizaje Extendido

Hasta que Simón diga

Sal con los estudiantes al patio y pídeles que formen una línea frente a ti. Asegúrate de que todos puedan verte. Menciona una pareja de “bucles hasta” como:

- Hasta que levante mi mano derecha, pueden caminar hacia mí • Hasta que diga “berenjena”, pueden caminar de espaldas hacia mí • Hasta que gire mi cabeza hacia la derecha, tienen que caminar como cangrejos

El primer estudiante en llegar a ti, gana. Si hay tiempo, haz que que otros tomen el rol de “Simón” frente a la clase.

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

► **AP** - Algoritmos y programación



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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

# Lección 19: Condicionales en Minecraft

Condicional | Minecraft

## Descripción

Esta lección da a los/as estudiantes la oportunidad de aprender y practicar los condicionales. Presenta personajes y escenarios de Minecraft, donde los/as estudiantes completan tareas tales como la minería y la construcción de estructuras usando sus programas.

## Propósito

Este conjunto de desafíos trabajará para solidificar y construir sobre el conocimiento de los condicionales y loops. Al combinar estos dos conceptos, los/as estudiantes podrán explorar el potencial para crear programas complejos e innovadores en un ambiente nuevo y emocionante.

## Agenda

### Actividad previa (15 mín)

Introducción

### Actividad Principal (30 min)

Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

### Actividad de Cierre (15 mín)

Escribir en el Diario

### Aprendizaje Extendido

### Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Definir las circunstancias en que ciertas partes de un programa deben ejecutarse y cuándo no.
- Determinar si se cumple un condicional según los criterios.

## Preparación

- ☐ Juegue **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web** correspondiente a este curso para encontrar áreas de potenciales problemas para su clase.
- ☐ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones**.
- ☐ Asegúrese que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones**.

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**
- Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones** [Hacer una copia](#)

Para los estudiantes

- Diario Think Spot - Diario de reflexiones** [Hacer una copia](#)

## Vocabulario

- **Condición** - una sentencia que un programa verifica si es verdadera o falsa. Si es verdadera, se realiza una acción. De lo contrario, se ignora.
- **Condicionales** - sentencias que solo se ejecutan bajo ciertas condiciones.

# Guía docente

## Actividad previa (15 mín)

### Introducción

Reúna a toda la clase y pida dos voluntarios/as para que caminen en línea recta hacia alguna dirección dentro de la sala. Si se encuentran una silla fuera de lugar deben pasar sobre ella, si llegan a una muralla deben sentarse.

Una vez que todos/as estén sentados/as, pregunte cómo programarían un robot para responder a una muralla o a una silla. Recuérdele que no pueden decir simplemente “Pasa sobre la silla”, a menos que ustedes sepan que hay una silla, y no siempre sabrán que hay una silla. Podría ser útil traducir la tarea en instrucciones como:

- Mientras haya un camino hacia adelante
  - Caminar hacia adelante.
  - Si hay una silla, pasar sobre ella.
- Sentarse

Diga a los/as estudiantes que estarán usando los *condicionales* para resolver este problema en Code.org. Dé las definiciones de:

- **Condición:** Una afirmación que un programa revisa si es verdadera o falsa. Si es verdadera se toma una acción. De lo contrario, la acción es ignorada.
- **Condicionales:** Afirmaciones que sólo se ejecutan bajo ciertas condiciones.

Abra una discusión de cuándo deberían usar un condicional en sus códigos.

## Actividad Principal (30 min)

### Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

A los estudiantes les va a encantar esta lección. Es probable que la mayoría de ellos hayan oído hablar de Minecraft, pero puedes hacer una breve introducción para aquellos que no lo conocen.

Minecraft es un juego de cubos. Puedes jugar como Alex o Steve mientras superas los laberintos. Tendrás que evitar la lava, recoger objetos y explorar un mundo hecho de cubos de cosas.

Pregunta a los estudiantes si alguna vez han jugado a Minecraft. Si ninguno lo ha hecho, pasa a la actividad principal. Si algunos lo han hecho, pídeles que expliquen el juego al resto de la clase. Si todos han jugado, sigue adelante y pasa a los desafíos en línea.

## Actividad de Cierre (15 mín)

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil y cómo se sienten. Puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy, y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el Diario:

- ¿De qué se trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- ¿Qué disfrutaste de los desafíos de hoy?
- ¿Cuándo utilizaste los condicionales en esta lección? ¿por qué los usaste?

## Aprendizaje Extendido



Más Minecraft

Si encuentra que su clase realmente goza del entorno de Minecraft, **aquí hay algunos enlaces a otros juegos de Minecraft que pueden jugar en línea**. Estos juegos también enseñarán habilidades básicas de codificación.

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

► **AP** - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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

# Lección 20: Condicionales & Loops en Cosechadora

Condicional | Bucle | Cosechadora

## Descripción

Los/as estudiantes practicarán bucles `mientras`, bucles `hasta`, y las declaraciones `si / si no`. Todos estos bloques usarán condicionales. Al practicar los tres, los/as estudiantes aprenderán a escribir código complejo y flexible.

## Propósito

Practicar el uso de condicionales en diferentes escenarios ayuda a desarrollar la comprensión de un/a estudiante de lo que puede hacer. En la lección anterior, los/as estudiantes sólo usaron condicionales para moverse alrededor de un laberinto. En esta lección, los/as estudiantes usarán condicionales para ayudar al agricultor a saber cuándo cosechar cultivos. Los nuevos patrones surgirán y los estudiantes usarán creatividad y pensamiento lógico para determinar las condiciones en las que el código debe ejecutarse y repetirse.

## Agenda

### Actividad previa (5 min)

Introducción

### Actividad Principal (30 min)

Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

### Actividad de Cierre (15 min)

Escribir en el Diario

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Anidar condicionales para analizar condiciones de valor múltiple mediante la lógica, `si`, `si no si`, `si no`.
- Integrar un bucle y una sentencia condicional.

## Preparación

☐ Juegue **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**

para encontrar áreas de potenciales problemas para su clase.

☐ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones.**

☐ Asegúrese que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones.**

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones** [Hacer una copia ▾](#)

Para los estudiantes

- **Diario Think Spot - Diario de reflexiones** [Hacer una copia ▾](#)

## Vocabulario

- **Condición** - una sentencia que un programa verifica si es verdadera o falsa. Si es verdadera, se realiza una acción. De lo contrario, se ignora.
- **Condicionales** - sentencias que solo se ejecutan bajo ciertas condiciones.
- **Bucle** - la acción de hacer algo una y otra vez.
- **Repetir** - hacer algo de nuevo.
- **Bucles mientras** - un bucle que se sigue repitiendo mientras una condición es verdadera.

# Guía docente

## Actividad previa (5 min)

### Introducción

Los estudiantes no deberían necesitar una introducción a los conceptos de hoy, pues han practicado con ellos en la lección previa. En lugar de eso, puedes compartir la historia de la cosechadora.

La cosechadora trata de cosechar cultivos como las calabazas, la lechuga y el maíz. Sin embargo, la granjera ha olvidado dónde plantó estos cultivos, así que necesita revisar cada planta antes de cosechar.

## Actividad Principal (30 min)

### Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

Los estudiantes continuarán trabajando con sentencias `si / si no`, bucles `mientras` y bucles `hasta`. Como estos desafíos son un poco más complejos, anima a los estudiantes a seguirlos hasta que sean capaces de describir lo que debe suceder con cada programa.

## Actividad de Cierre (15 mín)

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil, y cómo se sienten. Puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- ¿Cómo puedes ver que los condicionales son útiles en los programas?
- ¿Qué pasaría si la gente sólo hablara en declaraciones de `si/si no`? ¿Cuáles serían algunas de las ventajas y desventajas de esto?

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

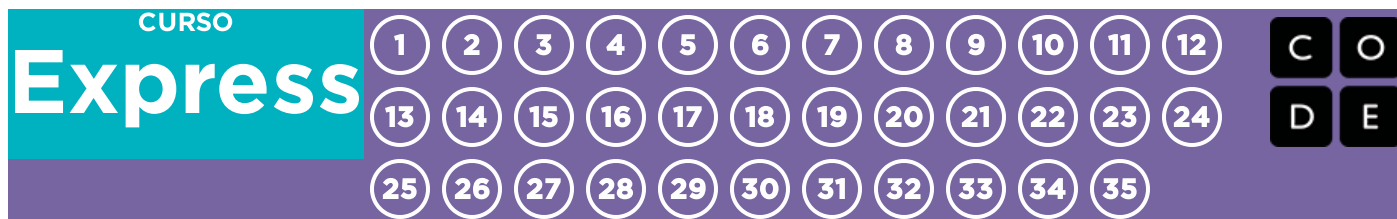
- **AP** - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



# Lección 21: Variables: Variables en Sobres

Sin conexión | Variable

## Descripción

Las variables se usan como parámetros de ubicación para valores como números y palabras. Las variables entregan mucha libertad en programación. En lugar de tener que escribir una frase muchas veces o recordar un número desconocido, los científicos de la computación pueden usar las variables para mencionarlos. Esta lección ayuda a explicar lo que son las variables y cómo podemos usarlas de diferentes formas. La idea de las variables no es un concepto fácil de entender, así que recomendamos dar suficiente tiempo para discutir al final de la lección.

## Propósito

Las variables son muy útiles en programación. Se le presentará este tema a los/as estudiantes usando sobres para representar a las variables a las que se les han dado nombres. El valor de la variable será escrito en una tarjeta dentro de un sobre. Esta lección ayuda a los/as estudiantes a comprender cómo los nombres pueden ser un parámetro de ubicación para valores en el mundo físico, así programar con variables parecerá menos confuso en el mundo virtual.

## Agenda

### Actividad previa (10 min)

Vocabulario  
Introducción

### Actividad Principal (20 min)

Variables de sobre - Ficha

### Actividad de Cierre (10 min)

Conversación Rápida: ¿Qué aprendimos?  
Escribir en el Diario

### Evaluación (10 min)

Variables de sobre - Evaluación

### Aprendizaje Extendido

## Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Identificar variables y determinar sus valores.
- Definir y llamar variables en el contexto de actividades de la vida real.
- Crear situaciones que requieren el uso de variables.

## Preparación

- ☐ Vea el **Variables de sobre - Video para el docente**.
- ☐ Consiga 6 o más sobres vacíos para la actividad previa y algunos más para la actividad principal.
- ☐ Imprima una **Variables de sobre - Ficha** por estudiante.
- ☐ Imprima una **Variables de sobre - Evaluación** para cada estudiante.
- ☐ Entregue a sus estudiantes sobres, papel, lápices y lápices mina.
- ☐ Asegúrese de que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones**.

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Variables en Sobres** - Video sin conexión ([descargar](#))
- **Variables de sobre** - Video para el docente
- **Variables de sobre** - Guía de respuestas en fichas [Hacer una copia](#)
- **Variables de sobre** - Guía de respuestas de la evaluación [Hacer una copia](#)

Para los estudiantes

- **Variables de sobre** - Ficha  
[Hacer una copia ▾](#)
- **Variables de sobre** - Evaluación  
[Hacer una copia ▾](#)
- **Diario Think Spot** - Diario de reflexiones  
[Hacer una copia ▾](#)

## Vocabulario

- **Variable** - un marcador de posición para una información que puede cambiar.

# Guía docente

## Actividad previa (10 min)

### Vocabulario

Esta lección tiene un término nuevo:

- **Variable** - Díganlo conmigo: Va-ria-ble

Un parámetro de ubicación para información que puede cambiar.

### Introducción

Pida cuatro voluntarios/as al frente de la sala y póngalos/as en fila. Infórmeles que van a escribir un poema para cada uno/a de ellos/as.

En la pizarra (o proyectado) escriba la oración para su primer estudiante (suponga que es Pedro):

"Mi estudiante Pedro, parado orgulloso  
, es un buen ejemplo para la gente"

Anime a los/as estudiantes a aplaudir por sus habilidades y agradézcale a Pedro por ser voluntario. Pídale a Pedro que vuelva a su puesto (o que vaya al final de la fila) mientras borra la pizarra, luego llame al siguiente voluntario/a (digamos que su nombre es Ana).

"Mi estudiante Ana, parada orgullosa  
, es un ejemplo para la gente"

Nuevamente, aceptando los aplausos, borre la pizarra e invite al/a siguiente voluntario/a.

"Mi estudiante Jenny, parada orgullosa  
, es un ejemplo para la gente"

Mientras llama al/a ultimo voluntario/a, pregunte si a todos en la clase les gustaría un poema escrito sobre ellos. ¿Quizás uno para todos en la escuela? ¡Cielos, eso va a tomar tiempo! Plantee esta pregunta a sus estudiantes:

"¿Cómo podría hacer esto más rápido?"

Sus estudiantes posiblemente se fijen en el hecho de que una palabra cambia, y es palabra es el nombre de la persona. Ayúdelos a ver la ubicación haciendo un círculo en el nombre Jenny en la pizarra y escriba "primerNombre" al lado.

"Tomaría mucho tiempo escribir un poema para todos en la escuela si no pudiera comenzar hasta saber de quién estoy escribiendo, ¿cierto?"

- ¿Cuánto tiempo crees que llevaría hacer video juego si no pudieran comenzar hasta que supieran tu nombre de usuario?
- ¿Qué tan caros serían los videos juegos si tuvieran que ser creados por separados para cada persona?
- ¿Cómo crees que podríamos evitarlo?

Para entonces, es muy probable que a su clase se le ocurra la idea de tener parámetro de ubicación. Con eso, están muy bien direccionados para comprender a dónde va esta lección.

- ¿Cómo llamaríamos a ese parámetro de ubicación?
  - Necesitamos ponerle un nombre que tenga sentido. No le daríamos el nombre "edad" si fuera un parámetro de ubicación para su nombre, ¿cierto?

Ahora, vamos a agregar más voluntarios/as. Dele a cada uno/a un trozo de papel para escribir sus nombres y pídale que lo pongan dentro de un sobre etiquetado "primerNombre".

Esta vez, ponga el poema en la pizarra con un espacio en blanco etiquetado "primerNombre" donde irá el nombre del/la estudiante.

- Pida al/la primer/a estudiante en la fila (posiblemente el/a último/a del ejemplo previo) que saque su nombre del sobre y eso es lo que usted escribirá en el espacio.
- Cuando borre la pizarra, solo borre el nombre del/a último/a estudiante.
- Llame al/a siguiente estudiante para mostrar su variable.
- Repita esto muchas veces mientras sea entretenido.

Llegó la hora de hacer la actividad principal.

## Actividad Principal (20 min)

### Variables de sobre - Ficha

Una vez que los/as estudiantes comprendan cómo los sobres se relacionan a las oraciones, entregue la guía de la actividad y deles tiempo para que preparen algunas variables propias.

#### Instrucciones:

- Divida a los/as estudiantes en grupos de 2 a 4.
- Pida a los/as estudiantes diseñar (dibujar) un robot.
- Después de 10-15 minutos, pida a los/as estudiantes que llenen sus sobres con detalles importantes de su robot como el nombre, la estatura, y el propósito.
- Recoja los sobres de cada grupo, luego llévelos al frente de la sala para compartir.
- Escriba en la pizarra: “El nombre de mi Robot es nombreRobot, mide númUnidad, y su propósito es propósito”.
- Use los sobres para llenar la variable apropiada en la oración, luego pida a cada grupo pararse cuando escuchen la oración que describe su creación.

## Actividad de Cierre (10 min)

### Conversación Rápida: ¿Qué aprendimos?

- ¿Qué aprendimos hoy?
- ¿Puedes pensar lugares dónde hayas visto variables antes?
- ¿Hay al menos una variable en la parte superior de la guía? ¿Puedes pensar cuál es?
- ¿Por qué crees que los profesionales no ponen espacios en las variables para nombres?
  - ¿Qué sucedería si hubiera una variable “ojo”, una variable “color” y una variable “color ojo”?
- Las variables también se pueden usar para almacenar números.
  - Suponga que tengo sobres etiquetados núm1 y núm2, y luego escribo núm1+núm2.
  - ¿Qué sucede si el sobre “núm1” contiene el número 4 y el “núm2” contiene el número 5?

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil y cómo se sienten, puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

#### Sugerencias para el diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- ¿Qué es una variable?
- ¿Por qué crees que las variables son importantes en programación?

## Evaluación (10 min)

### Variables de sobre - Evaluación

Dé tiempo suficiente a sus estudiantes para terminar esta evaluación. Si está dispuesto a pasar más tiempo en esta actividad, revise las respuestas con la clase.



# Aprendizaje Extendido

Use estas actividades para mejorar el aprendizaje de sus estudiantes. Pueden usarse como actividades fuera del aula u otros enriquecimientos.

## ¿Qué hay en el cuadrado?

- Dibuje cuadrados en un papel con operaciones matemáticas simples entre ellos.
  - Por ejemplo  $[] + [] = []$
- Tenga cuadrados de tamaños similares con números entre 1 y 20.
  - Pida a uno/a de los/as estudiantes crear una ecuación real, usando los número proporcionados.
- Una vez que el/a estudiante ha finalizado (y la clase ha verificado la ecuación) cambie uno de los números por otro, luego quite un segundo número completamente.
  - Diga a los/as estudiantes que hay un número escondido en el cuadrado vacío que hace que la ecuación esté correcta nuevamente.
  - ¿Qué número hay en el cuadrado?
- Haga este juego varias veces hasta que pueda quitar el número de cualquier ubicación y los/as estudiantes puedan descubrir cuál se supone que es.

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

- **AP** - Algoritmos y programación



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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

# Lección 22: Variables en Artista

Variable | Artista

## Descripción

En esta lección, los/as estudiantes explorarán la creación de diseños repetitivos utilizando variables del entorno del Artista. Los/as estudiantes aprenderán cómo se pueden usar las variables para hacer que el programa sea más fácil de escribir y más fácil de leer. Después de los desafíos guiados, los/as estudiantes terminarán en un nivel de juego libre para mostrar lo que han aprendido y crear sus propios diseños.

## Propósito

Las variables son esencialmente marcadores de posición para valores desconocidos al momento de ejecutar el programa o para valores que pueden cambiar durante la ejecución de este. Estas construcciones son vitales para crear código dinámico porque permiten que su programa cambie y crezca basado en cualquier número de modificaciones potenciales. Esta etapa enseña a los/as estudiantes qué son las variables, utilizando las capacidades más básicas de ajuste y utilización.

## Agenda

### Actividad previa (15 mín)

Introducción

### Actividad Puente - Variables (15 mín)

Actividad Sin Conexión usando Bloques de Papel  
Previsualización de los Desafíos Online con la clase

### Actividad Principal (30 mín)

Curso Rápido Fundamentos de Ciencias de la Computación  
- Sitio web

### Actividad de Cierre (15 mín)

Escribir en el Diario

## Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Asignar valores a las variables existentes.
- Utilizar variables en lugar de valores repetitivos dentro de un programa.
- Usar variables para cambiar valores dentro de un bucle.

## Preparación

- ▢ Juegue **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web** correspondiente a este curso para encontrar áreas de potenciales problemas para su clase.
- ▢ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones.**
- ▢ Asegúrese que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones.**

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones** [Hacer una copia ▾](#)

Para los estudiantes

- **Variables - Video para el estudiante (descargar)**
- **Bloques Blockly Sin Conexión (Cursos 2-5) - Manipuladores**
- **Diario Think Spot - Diario de reflexiones** [Hacer una copia ▾](#)

# Vocabulario

- **Variable** - un marcador de posición para una información que puede cambiar.

# Guía docente

## Actividad previa (15 mín)

### Introducción

Esta es la primera lección online que trata sobre variables, por lo que podría ser un ejercicio útil revisar la actividad sin conexión "Variables en Sobre", así como el vocabulario que se introdujo en esa lección.

- ¿Qué es una variable? (Un marcador de posición para una información que puede cambiar).
- ¿Cuándo puede ser útil una variable? (Cuando usted no sabe qué información se va a utilizar en un lugar determinado hasta el tiempo de ejecución, o cuando tiene muchos lugares en los que se va a utilizar una pieza de información, pero esa información podría cambiar algún día).

Pregunte a la clase cuándo podrían ver que una variable es útil para la programación. ¿Cuándo NO querrían usar una variable?

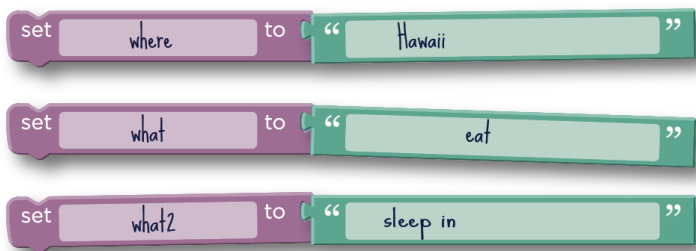
Si la clase parece interesada, continúe la discusión. De lo contrario, pase a una de las Actividades Puente.

## Actividad Puente - Variables (15 mín)

Esta actividad ayudará a llevar los conceptos sin conexión de "Variables en Sobre" al mundo en línea al que se están moviendo los/as estudiantes. Elija *una* de las siguientes opciones para realizar con su clase:

### Actividad Sin Conexión usando Bloques de Papel

Comience con una oración o párrafo en la pizarra que contenga por lo menos uno (y hasta cinco) espacios en blanco con nombre. Entregue bloques de papel a cada grupo de estudiantes. Desafíe a cada grupo a "establecer" sus variables para cada espacio en blanco. Llame a los diferentes grupos de estudiantes para compartir su tarea de cada variable y ver qué pasa.



"My favorite place to vacation is where ,  
because you can what all you like  
some times you even get to what !

Ahora, cambie la oración a una ecuación matemática. ¿Qué pasa a la oración " $X + Y =$ " cuando los/as estudiantes le asignan valores diferentes a las variables X & Y?

### Previsualización de los Desafíos Online con la clase

Muestre un desafío a la clase. Recomendamos el tercero. Construya el código de la manera larga primero (use números exactos para cada valor, en lugar de utilizar variables) y luego sugiera que debería intentar hacer los cuadrados de sólo 50 pixeles. ¡Qué terrible! ¿Qué han aprendido los/as estudiantes que les permita darle un

nombre a algo y usarlo tantas veces como quieran más adelante en el programa? Retroceda y añada una variable al principio. Poner la variable en 80, y sustituir todas las ocurrencias de 80 en el programa. Luego, cámbielo a 50. ¡Eso fue fácil!

## Actividad Principal (30 min)

### Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

Observa que esta etapa cubre primero la idea de una variable como una constante (una variable que se usa en muchos lugares, pero que no cambia). Una vez que se ha presentado esa idea, se voltea para mostrar cómo se puede incluir una variable para la información que cambia después de ejecutar el programa.

Cuidado con el desafío número 5. Es la primera vez que se espera que los estudiantes establezcan una variable por sí mismos. Esto puede ser difícil si no tienen un verdadero conocimiento del concepto. Si tienen problemas, envíalos de vuelta al nivel de predicción (n.º 4) y pídeles que expliquen a sus parejas por qué la respuesta terminó siendo la misma. Una vez que ambos miembros de la pareja estén convencidos, permíteles continuar con el desafío n.º 5.

## Actividad de Cierre (15 mín)

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil y cómo se sienten, puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el Diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- ¿Qué es una variable? ¿Por qué es útil en la programación?
- ¿Qué tan bien crees que entiendes las variables? (Responde en una escala del 1 al 5 o con un emoticón.) Si tienes problemas, ¿puedes poner en palabras lo que no entiendes?

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

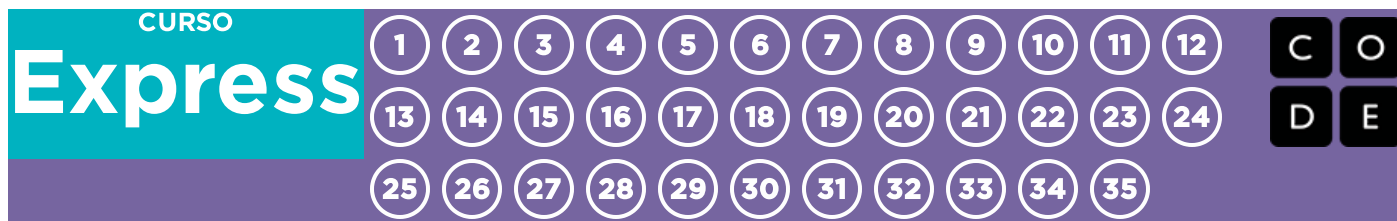
- AP - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



# Lección 23: Variables en Play Lab

Variable | Laboratorio de juegos

## Descripción

Los/as estudiantes podrán seguir practicando con las variables de esta lección creando escenas en Play Lab. Los/as estudiantes trabajarán con la entrada del usuario para establecer los valores de sus variables, y luego tendrán espacio para crear su propio mini-proyecto con variables.

## Propósito

Esta lección permite a los/as estudiantes utilizar variables para mostrar frases o conversaciones basadas en lo indicado por el usuario. Esta lección sirve como un maravilloso ejercicio de práctica de variables en la programación, ¡con una dosis extra de creatividad! Al final de la secuencia del desafío, los/as estudiantes tendrán la oportunidad de compartir sus proyectos con familiares y amigos/as.

## Agenda

### Actividad previa (15 mín)

Introducción

### Actividad Principal (30 min)

Curso Rápido Fundamentos de Ciencias de la Computación  
- Sitio web

### Actividad de Cierre (15 mín)

Escribir en el Diario

### Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Usar variables para contener palabras y frases.
- Usar variables junto con las sugerencias de caracteres.

## Preparación

- ▢ Juegue **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web** correspondiente a este curso para encontrar áreas de potenciales problemas para su clase.
- ▢ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones.**
- ▢ Asegúrese que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones.**

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones** [Hacer una copia ▾](#)

Para los estudiantes

- **Diario Think Spot - Diario de reflexiones** [Hacer una copia ▾](#)

## Vocabulario

- **Variable** - un marcador de posición para una información que puede cambiar.

# Guía docente

## Actividad previa (15 mín)

### Introducción

Pida a los/as estudiantes que levanten la mano y digan cuál es su comida favorita. Elija un par de estudiantes para que respondan. En una pizarra o cartulina, escriba:

A (nombre estudiante) le gusta (comida favorita)

Ejemplo: A Kiki le gusta la pasta.

Después de un par de estudiantes, pregunte a la clase si usted podría usar variables para continuar con esto el resto de la clase. Pregúnteles donde podría ir las variables y como las podrían llamar. Una vez que la discusión haya terminado, escriba:

[name] le gusta [food]

en la misma muestra

Pregunte a los/as estudiantes cuál es variable y cuál no. ¿Cómo lo saben? ¿Qué más podrían usar en oraciones como esta? (ejemplos: color favorito, ciudad, número de hermanos, etc.)

## Actividad Principal (30 min)

### Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

Tus estudiantes ya han conocido las variables, pero si piensas que están complicados con el concepto, recuérdales que le pregunten a sus compañeros antes de acercarse al profesor para que les ayude. Esto estimula la discusión y motiva una comunidad de aprendizaje.

#### 💡 Teacher Tip

Recuérdales a tus estudiantes compartir su trabajo sólo con sus amigos cercanos y familiares. Para mayor información, vea o muestre a la clase **Detente y Piensa Online - Video**.

## Actividad de Cierre (15 mín)

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil, y cómo se sienten. Puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- ¿Cuáles son algunas formas en las que has usados las variables hasta ahora?
- ¿Qué más crees que puedes hacer con las variables?

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

- **AP** - Algoritmos y programación

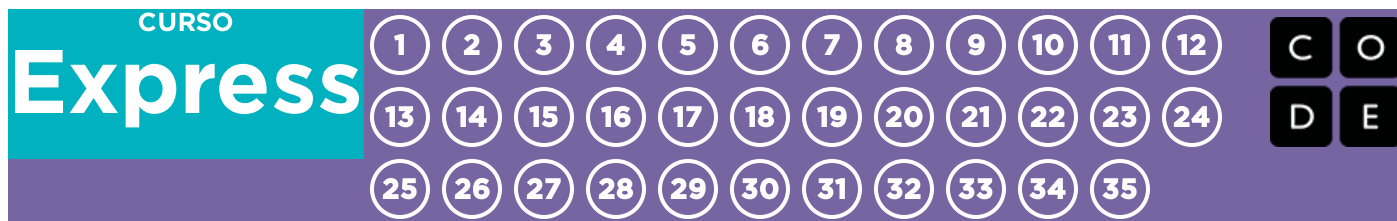


This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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





# Lección 24: Loops Para: Diversión con Loops Para

Sin conexión | Bucles desde

## Descripción

Sabemos que los loops nos permiten hacer cosas de forma repetida, pero ahora vamos a aprender cómo usar loops que tienen estructuras extras construidas en su interior. Estas estructuras nuevas permitirán a los/as estudiantes crear códigos que son más poderosos y dinámicos.

## Propósito

En este punto, los/as estudiantes son expertos/as en loops. Hoy, aprenderán acerca de otro loop usado comúnmente en programación. El Loop Para repite los comandos un cierto número de veces, pero también llevan registro de los valores que están repitiendo. Por ejemplo, un Loop Para que comienza en 4, termina en 8 y tiene un valor de paso de 1, repetirá 4 veces, pero los valores 4, 5, 6 y 7 también serán guardados para usarlos en otros lugares. Usando esta estructura con variables, se pueden crear programas fantásticos. Hoy, los/as estudiantes aprenderán los elementos básicos de los Loops Para, antes de sumergirse en la programación con ellos!

## Agenda

### Actividad previa (20 min)

Para Uno y Para Todos

### Actividad Principal (20 min)

Loops Para: Diversión con Loops Para - Ficha

### Actividad de Cierre (15 min)

Conversación Rápida: ¿Qué aprendimos?

Escribir en el Diario

### Evaluación (5 min)

Loops Para: Diversión con Loops Para - Evaluación

### Aprendizaje Extendido

## Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Determine el valor inicial, el valor final y el valor de paso para un bucle `desde`.
- Ilustrar los valores del contador que se alcanzan cada vez a través de un bucle desde durante el tiempo de ejecución.

## Preparación

- ☐ Vea **Loops Para: Diversión con Loops Para - Video para el docente**.
- ☐ Vea **Loops Para: Diversión con Loops Para - Video de lección en acción**.
- ☐ Imprima una **Loops Para: Diversión con Loops Para - Ficha** por grupo.
- ☐ Imprima una **Loops Para: Diversión con Loops Para - Evaluación** para cada estudiante.
- ☐ Asegúrese de que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones**.

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Loops Para: Diversión con Loops Para - Video sin conexión** ([descargar](#))
- **Loops Para: Diversión con Loops Para - Video para el docente**
- **Loops Para: Diversión con Loops Para - Video de lección en acción**
- **Diversión con Loop Para - Guía de respuestas en fichas** [Hacer una copia](#)

Para los estudiantes

- **Loops Para: Diversión con Loops Para** -  
Ficha Hacer una copia ▾

## Vocabulario

- **Bucles desde** - bucles que tienen un comienzo, un final y un incremento predeterminados (intervalo de paso).

# Guía docente

## Actividad previa (20 min)

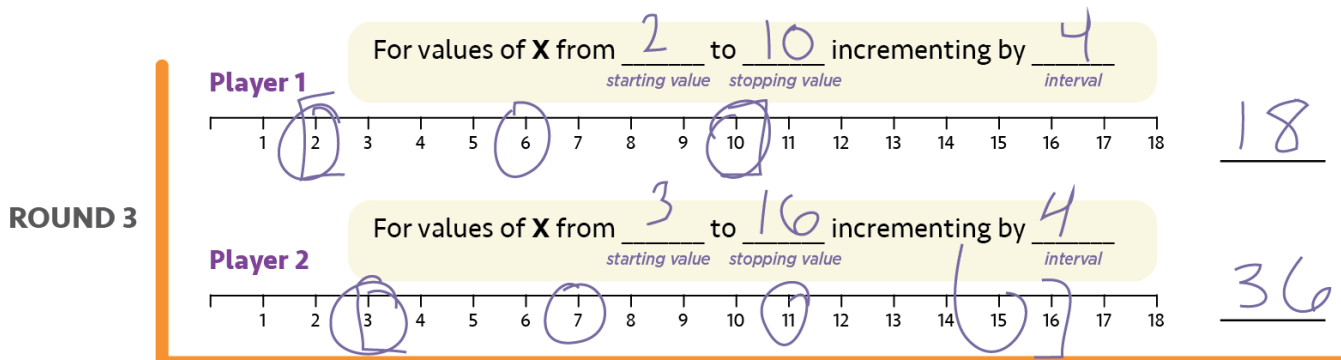
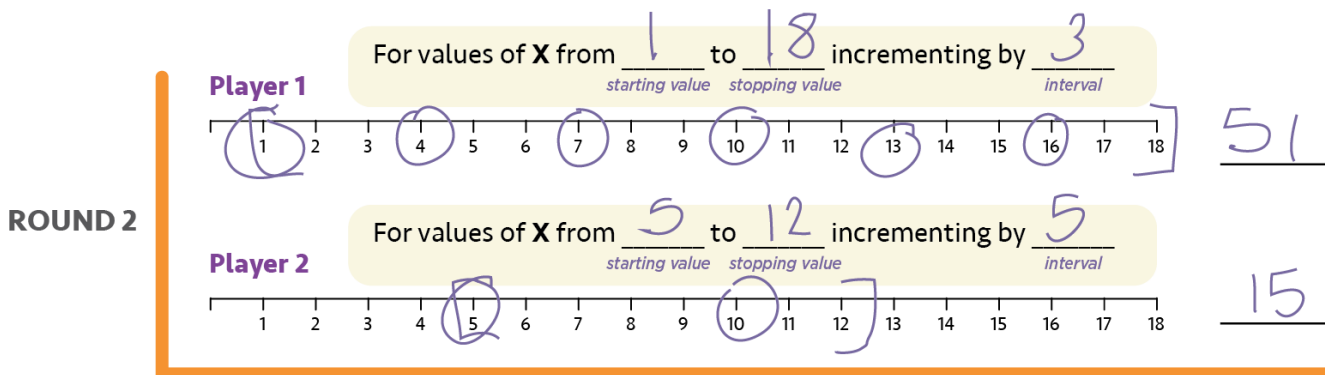
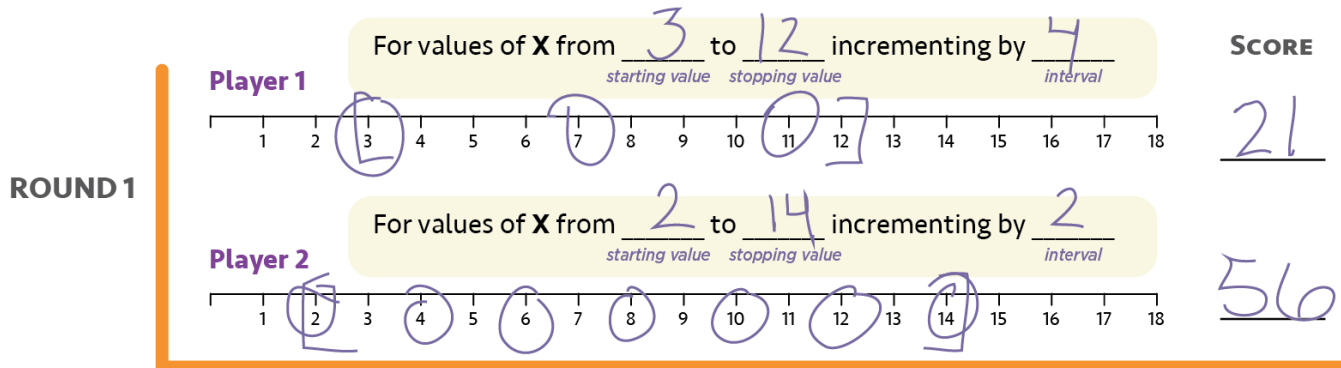
- **Loop Para** - Díganlo conmigo: Lup pa-ra

Son loops que tienen un comienzo, un final y un incremento (intervalo en pasos) predeterminado.

### Para Uno y Para Todos

- Señale que hay ciertos loops que son muy frecuentes, por ejemplo, los loops donde necesita llevar un registro de cuántas veces se ha usado
  - A veces, no quieres comenzar en uno.
  - A veces, no quieres contar de a uno.
  - Los loops "Para" te dan una forma poderosa de tener un contador que comienza cuando quieres, termina cuando quieres y se incrementa en cualquier tamaño que tú quieras.

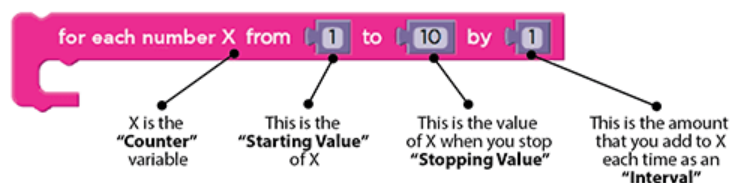
Aquí, puedes saltar a un ejemplo del juego



# Actividad Principal (20 min)

## Loops Para: Diversión con Loops Para - Ficha

A veces queremos repetir las cosas un cierto número de veces y tener un seguimiento de los valores a medida que lo hacemos. Para ello es útil el bucle desde. Cuando usas este bucle, sabes desde el principio cuál es tu valor inicial, cuál es tu valor final y cuánto cambia el valor cada vez durante el bucle.



Bloque de bucle desde (en inglés)

### Instrucciones:

- Divide a los estudiantes en parejas
- Para iniciar la ronda, cada estudiante lanza los dados tres veces:
  - Un dado para determinar el valor inicial de X
  - Tres dados para determinar el valor de detención de X
  - Un dado para determinar el valor de paso de X cada vez
- Usa una de las líneas numéricas proporcionadas para rastrear el bucle desde que han creado
  - Comienza en el valor inicial de X
  - Cuenta la línea numérica hacia atrás y haz un círculo en los números en el intervalo correspondiente
  - Detente cuando llegues al valor de detención predeterminado
- Agrega todos los valores marcados a su puntuación, y luego pasa el turno a otro jugador
- El que logre 2 de 3 gana

### Lesson Tip

Cuando juegas este juego, es como si ejecutases un bucle como este:

```
for (x=startValue; x <= stopValue; x = x + step){  
  circle currentXvalue;  
  add currentXvalue to roundScore;  
}
```

Para los estudiantes más jóvenes puede ser difícil entender esto si está escrito en pseudocódigo, pero resulta útil que expliques en voz alta (y quizás con un diagrama) lo que usarán como contenido de un bucle desde.

## Actividad de Cierre (15 mín)

### Conversación Rápida: ¿Qué aprendimos?

- ¿Qué intervalo necesitarías para contar desde 4 hasta 13 de tres en tres?
- ¿Qué tipos de cosas se te ocurren que podrías hacer con un Loop Para?
- ¿Puedes reproducir un loop normal usando un Loop Para?
- ¿Qué necesitarías hacer?

### Sugerencia para la lección

Las preguntas de la conversación Rápida tienen el objetivo de ver la imagen completa y como se relaciona con el mundo real del/a estudiante y su futuro. Use el conocimiento que tiene de su curso para decidir si hace esto como clase, en grupos o con el/a compañero/a de puesto.

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil y cómo se sienten, puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

#### Sugerencias para el Diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?

- ¿Qué es un Loop "Para"?
- ¿Para qué usarías un Loop "Para" en lugar de un Loop "Repetir" o un Loop "Mientras"?

## Evaluación (5 min)

### Loops Para: Diversión con Loops Para - Evaluación

Distribuya la hoja de la evaluación y permita que los/as estudiantes completen la actividad en forma independiente luego que las instrucciones queden claras. Esto debería serles familiar debido a las actividades en lecciones previas.

## Aprendizaje Extendido

Use estas actividades para mejorar el aprendizaje de sus estudiantes. Pueden usarse como actividades fuera del aula u otros enriquecimientos.

### Ejecutarlo al Revés

- Intente esta actividad nuevamente, pero esta vez pida a los/as estudiantes comenzar con un número seleccionado usando 3 dados, y detenerse en un número seleccionado por un dado. ¡Asegúrese de tener un incremento negativo!

### El avión (o el luce)

- Usando tiza, dibuje un avión (o luce) en el patio
  - Enumere los cuadrados desde el inferior al superior.
  - Pida a los/as estudiantes darse un cuadrado de inicio mutuamente, un cuadrado de detención, y cuántos necesitan saltar cada vez.
  - Cuando el participante termine, pídale que escriba el loop que han hecho.
  - Comience a agregar actividades adicionales para hacer en cada cuadrado, esto también le agregará complejidad a la parte escrita

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

- **AP** - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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

# Lección 25: Loops Para en Abeja

Bucles desde | Abeja

## Descripción

Con Abeja, esta lección se centra en los bucles "para" y utiliza una variable gradual para resolver desafíos más complicados. Los/as estudiantes comenzarán repasando los loops de las lecciones anteriores. Luego, repasarán una introducción a los loops "para", para que puedan resolver problemas complicados de manera más efectiva.

## Propósito

El concepto actual, bucles para, es un tema muy importante en la informática. No sólo son ampliamente utilizados, el proceso de aprendizaje para los loops para mejora el aprendizaje de otros conceptos importantes (tales como las variables y parámetros). Los/as estudiantes tendrán mucha práctica en el pensamiento crítico de los problemas al determinar los valores de inicio, final y pasos para cada loop para. Este concepto también utiliza mucho las matemáticas, así que siéntase libre de combinarlo con una lección de matemáticas para una experiencia de aprendizaje aún más profunda.

## Agenda

### Actividad previa (15 mín)

Introducción

### Actividad Puente - Loops Para (15 mín)

Actividad Sin Conexión Usando Bloques de Papel  
Pre visualización de los Desafíos en Línea con la Clase

### Actividad Principal (30 mín)

Curso Rápido Fundamentos de Ciencias de la Computación  
- Sitio web

### Actividad de Cierre (15 mín)

Escribir en el Diario

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Determine el valor inicial, el valor final y el valor de paso para un bucle `desde`.
- Reconocer cuándo usar un bucle `desde` y cuándo usar otros bucles, como `repetir` y `mientras`.

## Preparación

- ▢ Juegue **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web** correspondiente a este curso para encontrar áreas de potenciales problemas para su clase.
- ▢ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones.**
- ▢ Asegúrese que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones.**

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones** [Hacer una copia](#)

Para los estudiantes

- **Bloques Blockly Sin Conexión (Cursos 2-5)**  
- Manipuladores
- **Diario Think Spot - Diario de reflexiones**  
[Hacer una copia](#)

## Vocabulario

- **Bucles desde** - bucles que tienen un comienzo, un final y un incremento predeterminados (intervalo de paso).

# Guía docente

## Actividad previa (15 mín)

### Introducción

Recuerde a los/as estudiantes el trabajo que hicieron en "Loop Para Divertido". Abra una discusión sobre lo que aprendieron, por qué creen que podría ser útil y si se divertieron. Aquí hay algunas ideas para empezar la discusión.

- ¿Qué aprendiste en "Loop Para Divertido"?
- ¿Cuáles son los tres componentes principales de un Loop?
  - Valor inicial, intervalo de paso, valor final.
- ¿Por qué crees que un loop podría ser útil en la programación?
  - Es posible que muchos/as estudiantes no sepan la respuesta a esto. Deje que formulen hipótesis, pero no se detenga en esta pregunta por mucho tiempo.
- ¿Te divertiste aprendiendo sobre los loops? ¿Por qué si, o por qué no?
- ¿Estás emocionado/a por utilizar loops en desafíos en línea?

## Actividad Puente - Loops Para (15 mín)

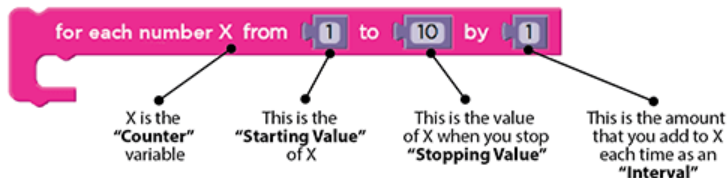
Esta actividad presentará los conceptos sin conexión de "Loop Para Divertido", al mundo en línea al que los/as estudiantes se están moviendo. Elija *una* de las siguientes actividades para realizar con su clase:

### Actividad Sin Conexión Usando Bloques de Papel

Divida a su clase en parejas. Idealmente forme las mismas parejas de la clase "Loop Para Divertido". Imprima un "Loop Para" de **Bloques Blockly Sin Conexión (Cursos 2-5) - Manipuladores** para cada pareja de estudiantes. Entregue uno a cada pareja. Haga que las parejas se alternen para tirar el dado para obtener los siguientes valores:

- Una tirada para el valor inicial.
- Cuatro tiradas para el valor final.
- Una tirada para el intervalo de paso.

Haga que cada pareja complete el "Loop Para" con el valor adecuado, en el espacio correcto.



Al usar una línea numérica básica, como la utilizada en "Loop Para Divertido", haga que los/as estudiantes marquen el inicio, final y valores intermedios que tocará el "Loop Para". Cuando todos hayan terminado, vea quién obtuvo la mayoría de los puntos sumando los números iniciales, medios y finales de cada pareja.

### Pre visualización de los Desafíos en Línea con la Clase

Muestre un desafío en línea de **Curso F Desafíos Online - 2018 - Sitio web** asociado a esta lección. Recomendamos el número nº 4, porque muestra la potencial solución y pide al/a estudiante que la evalúe.

Usando una línea numérica similar a las usadas en "Loop Para Divertido", marque los valores iniciales y finales del loop (si no está usando el desafío 4, necesitará encontrar una potencial solución primero). Con la ayuda de la clase, circule los valores entre el inicio y el final que tocará el "Loop Para". Si está trabajando en el desafío 4, pregunte a la clase cuál piensan que es la respuesta a la pregunta, dado lo que encontraron con la línea numérica.

## Actividad Principal (30 min)



## Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

Es posible que a algunos estudiantes les resulte difícil diferenciar entre los bucles `repetir` y los bucles `desde`. Te recomendamos tener un papel a mano para que los estudiantes hagan sus predicciones en relación a los valores de inicio, detención y paso. También puede ser útil implementar la programación en parejas.

## Actividad de Cierre (15 mín)

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil y cómo se sienten, puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el Diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- ¿En qué se diferencia un bucle `para` de un bucle `repetir`?
- ¿Por qué crees que los bucle `para` podrían ser útiles?

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

- **AP** - Algoritmos y programación



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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

# Lección 26: Loops Para en Artista

Bucles desde | Artista

## Descripción

En esta lección, los/as estudiantes continuarán practicando con los loop "para", pero esta vez con Artista. Los/as estudiantes completarán desafíos combinando las ideas de variables, loops y loops "para", para crear diseños complejos. Al final, tendrán la oportunidad de crear su propio arte en un nivel de juego libre.

## Propósito

La creatividad y el pensamiento crítico se conjugan a la perfección en esta lección. Los/as estudiantes continuarán su práctica con loops y variables mientras crean imágenes que los dejarán con la boca abierta. Esta lección inspira una mente creativa a la vez que enseña conceptos básicos de informática.

## Agenda

### Actividad previa (15 mín)

Introducción

### Actividad Principal (30 min)

Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

### Actividad de Cierre (15 mín)

Escribir en el Diario

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Usar bucles `desde` para cambiar el bucle varias veces con diferentes valores.
- Reconocer cuándo usar un bucle `desde` y cuándo usar otros bucles, como `repetir` y `mientras`.

## Preparación

- ☐ Juegue con **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web** asociada a esta lección para encontrar cualquier potencial área problemática para su clase.
- ☐ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones**.
- ☐ Asegúrese de cada estudiante tenga un **Diario Think Spot - Diario de reflexiones**.

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones** [Hacer una copia](#)

Para los estudiantes

- **Diario Think Spot - Diario de reflexiones** [Hacer una copia](#)

## Vocabulario

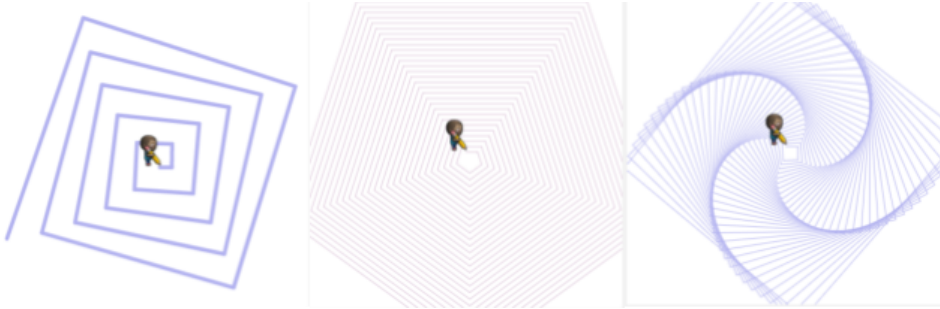
- **Bucles desde** - bucles que tienen un comienzo, un final y un incremento predeterminados (intervalo de paso).

# Guía docente

## Actividad previa (15 mín)

### Introducción

Dibuje en la pizarra para que toda la clase pueda ver (o utilizando un proyector) uno de los proyectos finales de **Curso F Desafíos Online - 2018 - Sitio web** asociado a esta lección. Recomendamos uno de los siguientes:



Pregunte a la clase como podría un computador hacer el dibujo que usted acaba de mostrar.

Luego de algunas cuantas respuestas, diga: ¡con los loops "Para", por supuesto!

Dígale a sus estudiantes que pronto aprenderán como crear estos geniales dibujos usando variables y loops "para".

## Actividad Principal (30 min)

### Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

Estos desafíos son muy divertidos, pero puede ser útil que los estudiantes tengan transportadores y papeles en blanco para ver estos diseños en un formato físico. Si no puedes aplicar esto en tu clase, intenta que los estudiantes sigan la pantalla de la computadora con sus dedos.

## Actividad de Cierre (15 mín)

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil y cómo se sienten, puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el Diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección?
- Dibuja uno de los diseños que hiciste hoy. ¿Cuál era el código necesario para crearlo?
- ¿Cuáles son algunos de los diseños que te gustaría crear? ¿Cómo crees que los loops "para" o las variables podrían ayudar a crearlos?

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

- **AP** - Algoritmos y programación



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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

# Lección 27: Funciones: composición de canciones con parámetros

Sin conexión | Función | Parámetro

## Descripción

Una de las estructuras más importantes en el mundo de la informática es la función. Las funciones (a veces llamadas procedimientos) son pequeños programas que usted puede usar una y otra vez dentro de un programa más grande. Esta lección ayudará a los estudiantes a comprender intuitivamente por qué combinar bloques de código en funciones, es una práctica tan útil, y cómo pueden usar esas estructuras incluso cuando los bloques de código son ligeramente diferentes.

## Propósito

El uso de las funciones ayuda a simplificar el código y desarrolla la capacidad del estudiante para organizar su programa. Los parámetros ayudarán a los/as estudiantes a personalizar sus funciones para que puedan ser utilizados para patrones que son similares, pero no idénticos. Los/las estudiantes reconocerán rápidamente que las funciones de escritura harán que sus programas largos sean más fáciles de leer y más fáciles de depurar, si algo sale mal.

## Agenda

### Actividad previa (15 mín)

Vocabulario

Cante una canción

### Actividad Principal (20 mín)

Funciones Sin conexión: Componer Canciones con Parámetros - Ficha

### Actividad de Cierre (15 mín)

Conversación Rápida: ¿Qué aprendimos?

Escribir en el Diario

### Evaluación (5 mín)

Funciones Sin conexión: Componer Canciones con Parámetros - Evaluación

### Aprendizaje Extendido

## Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Modificar funciones para aceptar parámetros.
- Describir cómo las funciones y los parámetros pueden hacer que los programas sean más fáciles de escribir.

## Preparación

- ▢Vea el **Componer Canciones y Componer Canciones con Parámetros (Funciones) - Video para el docente.**
- ▢Vea el **Funciones Sin conexión: Componer Canciones con Parámetros - Video de lección en acción.**
- ▢Imprima varios **Funciones Sin conexión: Componer Canciones con Parámetros - Ficha** para cada grupo.
- ▢Imprima una **Funciones Sin conexión: Componer Canciones con Parámetros - Evaluación** para cada estudiante.
- ▢Entre a internet, o descargue previamente canciones y letras para esta actividad.
- ▢Asegúrese de que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones.**

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Componer Canciones con Parámetros - Video sin conexión (descargar)**
- **Componer Canciones y Componer Canciones con Parámetros (Funciones) - Video para el docente**

- **Funciones Sin conexión: Componer Canciones con Parámetros** - Video de lección en acción
- **Funciones Sin conexión: Componer Canciones con Parámetros** - Ficha
- **Funciones Sin Conexión: Componer Canciones con Parámetros** - Guía de respuestas en fichas
- **Funciones Sin conexión: Componer Canciones con Parámetros** - Evaluación
- **Funciones Sin Conexión: Componer Canciones con Parámetros** - Guía de respuestas de la evaluación

Para los estudiantes

- **Diario Think Spot** - Diario de reflexiones

Hacer una copia ▼

## Vocabulario

- **Función** - un grupo de instrucciones de programación. Las funciones son abstracciones que se pueden volver a utilizar para reducir la complejidad a la hora de escribir y mantener programas.
- **Parámetro** - información adicional que se transfiere a una función para personalizarla para una necesidad específica.

# Guía docente

## Actividad previa (15 mín)

### Vocabulario

Esta lección tiene dos nuevas e importantes palabras:

- **Función** – Díganlo conmigo: Fun – ción.

Un bloque de código que se puede llamar fácilmente una y otra vez.

- **Parámetro** – Dilo conmigo: Pa – rá – me – tro.

Un bloque extra de información pasado a una función con el objeto de personalizarlo para una necesidad específica

### Cante una canción

- ¡Cuénteles a la clase que hoy es día de canción! Vamos a aprender una canción juntos/as.
  - Empiece con una canción fácil, escrita o proyectada en la pantalla.
  - Señale el estribillo y asegúrese de que la clase lo conozca antes de continuar con el resto de la canción.
  - Recorra rápidamente la canción, cantando con ellos en el comienzo y luego vea qué sucede cuando llegue a la parte que llama al estribillo.

Nota: La canción a continuación es una sugerencia. Usted puede elegir la canción que considere pertinente siempre y cuando esté alineada con los objetivos de la lección.



#### \*Chorus:

Little bunny Foo Foo  
Hopping through the forest  
Scooping up the field mice  
And bopping 'em on the head  
Down came the Fairy  
And she said  
"Little bunny Foo Foo  
I don't wanna see you  
Scooping up the field mice  
And bopping 'em on the head"\*

#### \*Song:

Chorus\*

*I'll give you 3 chances.  
Then I'll turn you into a goon!  
The next day. . .*

*Chorus*

*I'll give you 2 chances.  
Then I'll turn you into a goon!  
The next day. . .*

*Chorus*

*I'll give you 1 chance.  
Then I'll turn you into a goon!  
The next day. . .*

*Chorus*

#### 💡 Teaching Tip

Little Bunny Foo Foo is being used here as an example only. If your students know this song, feel free to use it. Otherwise, choose an appropriate song that they might be more familiar with (either from music class or the radio.)



*"I gave you two chances.  
Now I'll turn you into a goon!"  
(POOF!)  
And the moral of the story is:  
Hare today, goon tomorrow!*

- Es muy probable que la mayoría de la clase cante la letra del estribillo cuando señale esa parte.
  - Detenga la canción una vez que eso suceda, y destaque explícitamente lo que acaba de suceder.
    - Usted definió el estribillo.
    - Usted llamó el estribillo.
    - Ellos cantaron el estribillo.
- Pregúntele a la clase por qué suponen que usted sólo escribió el estribillo una vez en la parte superior del papel en lugar de escribirlo una y otra vez en cada lugar donde se supone que debe ser cantado.
  - ¿Cuáles son los otros beneficios de escribir el estribillo una sola vez cuando lo cantas muchas veces?

Ahora, imagine que esta canción es un programa de computador. Definir un título (como "estribillo") para un pequeño bloque de código que se usa una y otra vez, se llama crear una función. Esto es útil para los informáticos, por algunas de las mismas razones por las que es útil para los compositores.

- Ahorra tiempo al no tener que escribir todo el código una y otra vez en el programa.
- Si se equivoca, sólo debe cambiarlo en un lugar.
- El programa se siente menos complicado con las piezas que se repiten sólo una vez en la parte superior.

**¿Qué pasa con las canciones cuyo estribillo no es siempre exactamente igual? Puede igualmente usar un estribillo, pero debe tener una forma de decirle al cantante cuales son las palabras especiales que usará para cada verso.**

- Estas palabras especiales se llaman parámetros.
- En programación, los parámetros son instrucciones especiales para funciones como esta:

```
estribillo(parámetro1, parámetro2)
```

¿Siente que se está poniendo complicado? No se preocupe. Vamos a jugar un poco más con canciones para comprender de verdad como esta técnica es utilizada.

## Actividad Principal (20 min)

### Funciones Sin conexión: Componer Canciones con Parámetros - Ficha

Una manera fantástica de comparar las funciones con algo que vemos en nuestra vida diaria es mirar las canciones. Las canciones a menudo tienen ciertos grupos de letras que se repiten una y otra vez. A eso lo llamamos estribillo.

#### Instrucciones:

1. Dividir en grupos de 4, 5 o 6.
2. Entregue a cada grupo varias copias de la Guía para Componer Canciones.
3. Toque una breve canción para la clase que contenga un estribillo claro que no cambie de verso en verso.
4. Desafíe a la clase a identificar (y escribir) el estribillo.
5. Compare los resultados de cada grupo. ¿Todos escribieron lo mismo?
6. Intente la actividad nuevamente, pero esta vez con una canción que cambie durante cada repetición del estribillo. Algunos ejemplos buenos son: Old Mcdonald, Baby Bublebee o The Hokey Pokey.

Discuta con la clase:

- ¿Pueden los estudiantes identificar un estribillo cuando algunas palabras cambian?

#### 🔗 Recomendación para la lección

Para aumentar el interés, busque una canción popular en internet. Muéstrole a los/las estudiantes que el estándar para repetir la letra es lo que define el coro en la parte de arriba y lo llama desde el cuerpo de la canción.

- ¿Cómo podrían usar la misma idea de cantar un estribillo cuando los versos del estribillo son diferentes? éste es diferente de las estrofas
- Estas palabras y frases que cambian se llaman "parámetros" y se pueden pasar al estribillo de la siguiente manera: estribillo (cow, moo)
- Juegue este juego una y otra vez hasta que la clase tenga pocos problemas para identificar los estribillos.

A menudo es más fácil hacer que la clase escuche (o vea) la canción, y luego votar acerca de cuál es el estribillo cantando juntos, en lugar de escribirlo todo. Si elige este método, considere la posibilidad de que la clase haga un estribillo escrito para la selección final de canciones, para asegurarse de que los/as estudiantes visuales reciban el refuerzo adecuado.

#### 💡 Lesson Tip

Es más emocionante para los/as estudiantes hacer esta clase con música popular de la radio, pero si le está costando encontrar canciones apropiadas donde las letras se repitan de forma exacta, aquí hay algunas alternativas.

Nota: Las canciones Old McDonald, Baby Bumblebee o The Hokey Pokey son una sugerencia. Usted puede elegir la canción que considere pertinente, siempre y cuando esté alineada con los objetivos de la lección.

- **5 Little Monkeys**
- **Old MacDonald**
- **Hokey Pokey**
- **BINGO**
- **Baby Bumble Bee**

## Actividad de Cierre (15 mín)

### Conversación Rápida: ¿Qué aprendimos?

- ¿Preferirías escribir la letra de una canción muchas veces o definir un estribillo?
- ¿Crees que sea posible hacer varios estribillos para la misma canción?
- ¿Tiene sentido hacer un estribillo nuevo, cada vez que se necesita en una canción?

#### 💡 Lesson Tip

Las preguntas de la Charla Exprés buscan ver la foto completa para ayudar a pensar cómo la lección se relaciona con el mundo y el futuro de los/as estudiantes. Use su conocimiento de los/as estudiantes para decidir si quiere discutir esto como clase, en grupos o con el/a compañero/ de puesto.

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil y cómo se sienten. Puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy, y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el Diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- ¿Cómo ves que las funciones son útiles en la ciencia de la computación?
- Describe por qué los parámetros son útiles cuando se escribe la letra de una canción en la que el estribillo cambia ligeramente.

## Evaluación (5 min)

### Funciones Sin conexión: Componer Canciones con Parámetros - Evaluación

Distribuya la hoja de la evaluación y permita que los/as estudiantes completen la actividad en forma independiente luego que las instrucciones queden claras. Esto debería serles familiar debido a las actividades en lecciones previas.

## Aprendizaje Extendido

Usa estas actividades para mejorar el aprendizaje de los estudiantes. Pueden usarse como actividades fuera del aula o como otro tipo de enriquecimiento.

### Crear tu canción

- Para comenzar, creen todos juntos un estribillo, y luego repítanlo entre los versos de una canción creada específicamente.
- Hagan un cambio en el estribillo y evalúen cuanto más fácil es cambiarlo en un solo lugar.
- Cambien el estribillo nuevamente y háganlo más largo que el original.
- Agreguen un segundo estribillo que alternarán entre los versos.
- Agreguen parámetros a uno de los estribillos y vean cuántas opciones tienen.

### Componer un programa

- ¿Qué pasaría si actuamos las canciones en lugar de cantarlas? De repente, nuestros estribillos no serán palabras sino una función de acciones repetidas.
- Usa el concepto de las flechas de la lección de Programación con papel cuadriculado y crea un programa con muchas instrucciones repetidas.
  - Encierra en un círculo esas acciones repetidas, para que la clase pueda ver dónde están.
  - Define una función llamada "Estribillo" al inicio del programa.
  - Haz una cruz en los lugares en que aparezcan las acciones repetidas del programa y escribe "Estribillo" en su lugar.
- Repite hasta que la clase pueda realizar este proceso casi sin instrucciones.
- ¿Puedes descubrir cómo pasar parámetros en este ejercicio?

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

► **AP** - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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

# Lección 28: Funciones en Abeja

Función | Abeja

## Descripción

Esta lección enseña a los/as estudiantes cómo crear funciones simples usando nuestro sofisticado editor de funciones "modales", preparándoles el camino para que puedan incorporar parámetros en futuras lecciones.

## Propósito

Los/as estudiantes descubrirán la versatilidad de programar practicando las funciones en diferentes ambientes. Aquí, reconocerán patrones en el laberinto de la abeja. La abeja necesitará navegar el área de juego, recoger néctar y hacer miel. Aprenderán a organizar sus programas y a crear funciones para códigos repetidos.

## Agenda

### Actividad previa (15 mín)

Introducción

### Actividad Puente - Funciones (15 mín)

Actividades Sin Conexión Usando Bloques de Papel  
Previsualización de los Desafíos Online con la Clase

### Actividad Principal (30 min)

Curso Rápido Fundamentos de Ciencias de la Computación  
- Sitio web

### Actividad de Cierre (15 mín)

Escribir en el diario

### Aprendizaje Extendido

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Categorizar y generalizar código en funciones útiles.
- Reconocer cuándo una función podría ayudar a simplificar un programa.

## Preparación

- ▢ Juegue **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web** correspondiente a este curso para encontrar áreas de potenciales problemas para su clase.
- ▢ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones.**
- ▢ Asegúrese que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones.**

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones** [Hacer una copia ▾](#)

Para los estudiantes

- **Bloques Blockly Sin Conexión (Cursos 2-5)**  
- Manipuladores
- **Diario Think Spot - Diario de reflexiones**  
[Hacer una copia ▾](#)

## Vocabulario

- **Función** - un grupo de instrucciones de programación. Las funciones son abstracciones que se pueden volver a utilizar para reducir la complejidad a la hora de escribir y mantener programas.
- **Parámetro** - información adicional que se transfiere a una función para personalizarla para una necesidad específica.

# Guía docente

## Actividad previa (15 mín)

### Introducción

Los/as estudiantes que han hecho **Curso E Desafíos Online - Sitio web**, ya tienen experiencia en el uso de funciones para resolver desafíos en línea, pero el editor de funciones en este curso es ligeramente diferente. Informe a los/as estudiantes que realizarán una revisión rápida de las funciones simples antes de pasar a desafíos más difíciles con el nuevo editor "modal".

Para los/as estudiantes que están menos familiarizados con el uso de funciones en línea, empiece repasando las palabras del vocabulario de las "Funciones Sin Conexión": Composición de canciones con parámetros".

- **Función** - Díganlo conmigo: Fun-ción.

Un trozo de código al que puedes llamar una y otra vez.

- **Parámetro** - Díganlo conmigo: Pa-rá-me-tro.

Un pedazo extra de información que usted pasa a la función para personalizarla para una necesidad específica.

Dígale a la clase que el uso de funciones con parámetros tiene dos componentes principales.

1. **La Declaración:** Las declaraciones de función, son las que crean una función. En una declaración de función, se rellena la función con el código y se le da un nombre. Dentro de la declaración de función, se debe indicar dónde se utiliza el parámetro dentro del código de función. Debe declarar una función antes de poder utilizarla.
2. **El Llamado:** Las llamadas de función, son las que hacen que el programa ejecute el código de la función. Para llamar una función, coloque el nombre de la función en su programa con un valor para el parámetro. Asegúrese de que su función está correctamente definida (con un parámetro) antes de llamarla en su programa.

La clase puede usar la composición de canciones como ejemplo para entender estos dos componentes. En la actividad sin conexión, la función que contiene la letra del estribillo se llamaba "estribillo". Cuando hicimos esta función por primera vez, marcamos con un círculo la letra que iría en la función. Una vez que nombramos la función, pudimos leer las letras y reemplazar las letras repetidas del coro por una llamada de función a "estribillo".

Continúe la conversación hasta que los/as estudiantes tengan una comprensión básica de las funciones que se están explicando y llamando. Si los/as estudiantes no llegan a este punto, asegúrese de hacer una de las actividades de enlace antes de pasar a los desafíos de Code.org.

Nota: Los/as estudiantes no usarán parámetros en sus funciones hoy. Sin embargo, es bueno revisar para la próxima vez, qué son los parámetros y por qué se usan.

## Actividad Puente - Funciones (15 mín)

Esta actividad ayudará a ver los conceptos de "Funciones Sin Conexión: Composición de canciones con Parámetros" en el mundo en línea al que los/as estudiantes se están moviendo. Elija una de las siguientes opciones para su clase:

### Actividades Sin Conexión Usando Bloques de Papel

Escoja una canción que los/as estudiantes disfruten e imprima la letra. Puede utilizar la misma canción de "Funciones sin Conexión: Composición con Parámetros". Divida su clase en grupos o parejas. Reparta las letras impresas (incluyendo el estribillo repetido) y la función con bloques de parámetros de **Bloques Blockly Sin Conexión (Cursos 2-5) - Manipuladores** para cada grupo o pareja de estudiantes. Vea las recomendaciones para la lección, para más detalles.

Pida a los/as estudiantes que tachén cualquier parte de la canción que se pueda convertir en una función, incluso si tiene un par de palabras diferentes (el estribillo es un buen ejemplo) y que la pongan en los bloques de función proporcionados. Los/as estudiantes deben rellenar la declaración de función con un nombre de función y las palabras de las letras repetidas. Una vez hecha la declaración de función, pida que rellenen las llamadas de función y las coloquen encima de las letras tachadas.

Una vez que todos los grupos o parejas han terminado, pregunte a la clase donde ubicaron sus funciones y por qué. ¿Hicieron todos/as la misma función? ¿Qué tan a menudo se repite la función?

#### 💡 Sugerencia para la lección

Bloques de función:



El bloque a la izquierda es una función de declaración, un bloque que los/as estudiantes nombrarán y completarán en la función. El bloque a la derecha es la función llamada, un bloque que hace el código de función correr. Los/as estudiantes necesitarán múltiples bloques de la función llamada.

## Previsualización de los Desafíos Online con la Clase

Muestre un desafío de **Curso F Desafíos Online - 2018 - Sitio web**. Recomendamos el nº 12 para esta actividad. Con la clase, resuelva el desafío sin usar funciones. Una vez que haya obtenido la solución, muéstreala en una pizarra o en un proyector. Pida a la clase que señale el código repetido. Pregunte cómo simplificarían el programa. En la pizarra o en el proyector, reescriba el programa sin el código repetido, pero dejando una línea de espacio. En ese/esos espacio(s), llame una función. Declare la función como el bloque de ejemplo a la izquierda en la recomendación para la lección. Pregunte a la clase qué creen que hará el código ahora.

Inicie una discusión con la clase sobre por qué las funciones pueden ser útiles en la programación. Invite a los/as estudiantes a discutir la diferencia entre las funciones y los loops.

## Actividad Principal (30 min)

### Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

A los estudiantes podría resultarles útil escribir códigos sin funciones y, luego, crear funciones a partir de los códigos repetidos. Si no quieren hacer esto en el espacio de trabajo de Code.org, te recomendamos que les des papel y lápiz para que escriban (o dibujen) sus ideas.

## Actividad de Cierre (15 mín)

### Escribir en el diario

Hacer que los/as estudiantes escriban acerca de lo que han aprendido, por qué es útil, y cómo se sintieron puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy y a construir una hoja de revisión que pueden mirar en el futuro.

Sugerencias para el Diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- ¿Cómo ayudaron las funciones a la abeja a recolectar néctar y hacer miel de forma más eficiente?
- ¿Puedes imaginar usar parámetros en estos desafíos? Si es así, explica cómo. Si es no, ¿Por qué no?

## Aprendizaje Extendido

Dibujar con Funciones

Divida a la clase en grupos de 2-3 estudiantes. Haga que cada grupo escriba una función que dibuje algún tipo de forma y un programa que use esa función. Dependiendo de la creatividad o del enfoque de los grupos, los/as estudiantes pueden necesitar que se les asignen las formas a crear. Una vez que todos/as hayan terminado, pida a los grupos que intercambien programas. En una hoja distinta, cada grupo debe dibujar lo que crea el programa. Los grupos deben regresar los programas y los dibujos al grupo original.

¿Todos los grupos obtuvieron el dibujo que esperaban? Si no, ¿Qué salió mal? Haga que la clase realice el proceso de debugging y lo intenten de nuevo.

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

► **AP** - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



# Lección 29: Funciones con Parámetros en Artista

Función | Parámetro | Artista

## Descripción

En esta lección, los/as estudiantes continúan trabajando con funciones con y sin parámetros. Tendrán la oportunidad de crear sus propios dibujos, antes de modificar las funciones en un nivel de juego libre.

## Propósito

Esta lección ofrece un espacio para crear algo de lo que se sientan orgullosos/as.

Estos desafíos permiten a los/as estudiantes crear imágenes complejas a partir de proyectos anteriores más simples. Al final de esta lección, los/as estudiantes tendrán imágenes de las que sentirse orgullosos.

## Agenda

### Actividad previa (10 min)

Introducción

### Actividad Principal (30 min)

Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

### Actividad de Cierre (15 min)

Escribir en el Diario

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Practicar la abstracción con el uso de parámetros.
- Reconocer cuándo una función podría ayudar a simplificar un programa.

## Preparación

- ☐ Juegue **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web** para encontrar áreas de potenciales problemas para su clase.
- ☐ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones.**
- ☐ Asegúrese que cada estudiante tenga un **Diario Think Spot - Diario de reflexiones.**

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones** [Hacer una copia](#)

Para los estudiantes

- **Diario Think Spot - Diario de reflexiones** [Hacer una copia](#)

## Vocabulario

- **Función** - un grupo de instrucciones de programación. Las funciones son abstracciones que se pueden volver a utilizar para reducir la complejidad a la hora de escribir y mantener programas.
- **Parámetro** - información adicional que se transfiere a una función para personalizarla para una necesidad específica.

# Guía docente

## Actividad previa (10 min)

### Introducción

Pregunte a la clase si recuerdan haber usado los parámetros de "Funciones Sin Conexión": Composición de canciones con parámetros".

- ¿Qué significa "utilizar un parámetro" con funciones?
  - El uso de un parámetro significa que la función toma un valor de variable que puede especializar la función. En "Composición con Parámetros" esto significa que podemos cambiar las letras repetidas para que sean un poco diferentes cada vez.
- ¿Por qué usaríamos un parámetro con una función?
  - Utilizamos parámetros con funciones para no tener que escribir múltiples funciones que son muy similares. Si quisiéramos dibujar tres cuadrados con tres longitudes de lados diferentes, sólo tendríamos que escribir una función con un parámetro en lugar de tres funciones diferentes sin usar parámetros.
- ¿Por qué no usamos siempre parámetros con funciones?
  - No necesitamos SIEMPRE una función personalizable. A veces las funciones son sólo una forma práctica de reutilizar un código idéntico en múltiples lugares.

¡Dígale a la clase que harán dibujos increíbles en Artista, usando funciones con parámetros!

## Actividad Principal (30 min)

### Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

Pide a los estudiantes que cierren los ojos y levanten la mano. Si se sienten bien con el uso de los parámetros, pídeles que abran todos los dedos (como en "choca esos cinco"). De lo contrario, haz que levanten el puño. Si se sienten en algún punto intermedio, pídeles que levanten uno, dos, tres o cuatro dedos de la mano.

A partir de allí, determina si tu clase necesita practicar más las funciones antes de pasar a los desafíos en línea. Si solo una pequeña parte de tu clase no se siente cómoda con el uso de los parámetros, asegúrate de emplear la programación en parejas en esta lección.

## Actividad de Cierre (15 mín)

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil y cómo se sienten. Puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy, y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el Diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- Haz un dibujo que hiciste hoy. ¿Puedes escribir el código necesario para crear esto?
- Dibuja lo que te gustaría crear con código. Intenta escribir o redactar el código que haría ese dibujo.

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

► **AP** - Algoritmos y programación



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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

# Lección 30: Funciones con Parámetros en Abeja

Función | Parámetro | Abeja

## Descripción

Esta lección presenta el entorno de Abeja, y continúa con el concepto de funciones con parámetros de la anterior etapa Artista. Los/as estudiantes practicarán la escritura y el uso de funciones para seguir caminos complejos y recolectar patrones de néctar y miel.

## Propósito

Las funciones son increíblemente importantes en las ciencias de la computación por muchas razones. La capacidad de desglosar y categorizar el código se volverá inmensamente importante a medida que los programas que sus estudiantes escriben se vuelven más y más complejos. Las funciones con parámetros requieren un nivel extra de habilidad. El uso de funciones con parámetros enseña a sus estudiantes a reconocer cuándo se necesita una función y si esa función puede generalizarse lo suficiente para ser utilizada en múltiples casos. Esta lección, junto con las anteriores sobre funciones con parámetros, construye un sólido conjunto de habilidades de pensamiento crítico y resolución de problemas.

## Agenda

### Actividad previa (15 mín)

Introducción

### Actividad Principal (30 mín)

Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

### Actividad de Cierre (15 mín)

Escribir en el Diario

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Reconocer las tareas repetidas que deben especializarse caso por caso y crear funciones para ejecutar estas tareas de manera eficiente.
- Usar funciones predefinidas con parámetros para completar tareas que se repiten con frecuencia.

## Preparación

- ☐ Juegue **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web** asociado a este nivel, para encontrar cualquier potencial área problemática para su clase.
- ☐ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones.**
- ☐ Asegúrese de cada estudiante tenga un **Diario Think Spot - Diario de reflexiones.**

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web**
- **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones** [Hacer una copia ▾](#)

Para los estudiantes

- **Diario Think Spot - Diario de reflexiones** [Hacer una copia ▾](#)

# Vocabulario

- **Función** - un grupo de instrucciones de programación. Las funciones son abstracciones que se pueden volver a utilizar para reducir la complejidad a la hora de escribir y mantener programas.
- **Parámetro** - información adicional que se transfiere a una función para personalizarla para una necesidad específica.

# Guía docente

## Actividad previa (15 mín)

### Introducción

Antes de la clase, prepare un par de caminos en la sala de clases para que los/as estudiantes puedan caminar. Asegúrese de que el número de pasos sea obvio, ya sea usando cinta adhesiva o usando huellas de pies. La longitud de estos recorridos debe variar.

Reúna a la clase e indique que hay diferentes senderos para caminar, pero no quiere tener que escribir funciones separadas, para caminar por cada uno de ellos.

En cambio, en un lugar donde toda la clase pueda ver, escriba o muestre lo siguiente

**FUNCIÓN** - "trayectoria", **PARÁMETRO** - "paso"

- Repita la cantidad de "pasos":
  - Camine hacia adelante

Pregunte a la clase si saben lo que significa el código que usted escribió. Dígale a la clase que en lugar de escribir una función única para cada sendero, usted escribió una función que puede ser personalizada a la longitud de este.

Esto se hizo declarando una función, "trayectoria", y luego dándole un parámetro, "paso". La variable "paso" permite mantener el número de pasos de cada trayectoria.

Juegue con la función para cada sendero, teniendo a un/a voluntario/a que diga el número de pasos en un sendero y otro/a voluntario/a caminando por el sendero de acuerdo al código.

## Actividad Principal (30 min)

### Curso Rápido Fundamentos de Ciencias de la Computación - Sitio web

A medida que la clase trabaja con estos desafíos, pregunta lo siguiente a cada estudiante.

- ¿Estás usando una función? ¿Por qué? ¿Por qué no?
- Si no estás usando una función, ¿crees que sería útil aquí?
- Si estás usando una función, ¿empleas un parámetro? ¿Por qué? ¿Por qué no?
- Si no estás usando un parámetro, ¿crees que sería útil aquí? ¿Por qué? ¿Por qué no?

A veces, los estudiantes no necesitan una función o una función con parámetros, pero siempre deben saber *por qué* hacen lo que hacen.

## Actividad de Cierre (15 mín)

### Escribir en el Diario

Hacer que los/as estudiantes escriban acerca de lo que aprendieron, por qué es útil y cómo se sienten. Puede ayudar a solidificar cualquier conocimiento que hayan obtenido hoy, y a construir una hoja de revisión para que la revisen en el futuro.

Sugerencias para el Diario:

- ¿De qué trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- ¿Crees que los parámetros son útiles en el código?
- ¿Cuándo usó un parámetro y cómo cambió la forma en que escribió el resto de tu programa?

## Normas de aprendizaje



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



# Lección 31: Explorar Ideas de Proyectos

Proyecto | Definir | Preparar | Probar | Revisión | Reflexionar

## Descripción

Las próximas cinco lecciones ofrecen a los/as estudiantes la oportunidad de poner en práctica sus habilidades de codificación en un proyecto final. Este proyecto ayudará a las personas a adquirir experiencia con la codificación y a producir un ejemplo para compartir con sus compañeros/as y seres queridos. Se trata de un proyecto para varias lecciones o varias semanas en las que los/as estudiantes dedicarán tiempo a la lluvia de ideas, al aprendizaje del proceso de diseño, a la construcción y a la presentación de su trabajo final.

En la etapa "Explorar", los/as estudiantes jugarán con programas de Artista y Play Lab ya construidos para inspirarse. A continuación, los/as estudiantes aprenderán sobre el proceso de diseño y cómo implementarlo en sus propios proyectos. Luego se les dará el espacio para crear su propio proyecto en Artista, Play Lab, o cualquier otra interfaz que usted se sienta cómodo proporcionando. (Esta es probablemente la etapa más larga del proyecto.) Los/as estudiantes revisarán su código después de las pruebas y la revisión de los compañeros. Finalmente, los/as estudiantes podrán presentar sus trabajos terminados a sus compañeros/as de clase.

## Propósito

Explorar ideas de proyecto está hecho para inspirar a los/as estudiantes con ideas realistas y entretenidas para sus proyectos finales.

## Agenda

### Día 1 - Explora Ideas Para Proyectos (45 min)

Ejemplos de proyectos

### Día 2 - El Proceso de Diseño (45 min)

Definir y Preparar

### Día 3. Construye Tu Proyecto (45 min)

Intentar

### Día 4 (Recomendado para 5º Grado) - Revisa Tu Proyecto (45 min)

Reflexionar e Intentar nuevamente

### Día 5 & 6 - Presenta Tu Proyecto (45 min cada uno)

Presentaciones

### Actividad de Extensión

## Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Aprender a planificar con anticipación para una asignación continua.
- Poder explicar cómo las limitaciones del sistema pueden afectar el diseño del proyecto.
- Describir cómo el compromiso puede ayudar a mantener en marcha un proyecto e inspirar creatividad.

## Preparación

- ▢ Juegue online **Ejemplos de Proyectos Curso Rápido** para hacerse una idea de las fortalezas, debilidades y limitaciones de la herramienta.
- ▢ Decida si hará que sus estudiantes realicen o no la sección **Revisions** (recomendada para el Curso F).
- ▢ Imprima una copia de **Proceso de Diseño - Guía de preparación para el docente** para cada estudiante.
- ▢ Modifique **Proyecto Final de Fundamentos de Ciencias de la Computación - Rúbrica** para que sea acorde a los objetivos de su clase e imprima una copia para cada estudiante.
- ▢ Modifique **Diseño del Proyecto Final - Ficha** para que sea acorde a su clase e imprima un paquete para cada estudiante.

## Enlaces

**¡Importante!** Haz una copia de los documentos que piensas compartir con los estudiantes.

Para los docentes

- **Ejemplos de Proyectos Curso Rápido**
- **Proceso de Diseño - Guía de preparación para el docente** [Hacer una copia ▾](#)

- **Diseño del Proyecto Final** - Ficha  
[Hacer una copia ▼](#)
- **Proyecto Final de Fundamentos de Ciencias de la Computación** - Rúbrica
- **72 formas creativas de que tus estudiantes muestren lo que saben** - Sitio web
- **Google Slides**

## Vocabulario

- **Definir** - descubrir los detalles de los problemas que estás tratando de resolver.
- **Preparar** - investigar, planificar y adquirir materiales para la actividad que estás a punto de realizar.
- **Reflexionar** - pensar cuidadosamente en algo con la intención de mejorar el resultado en el futuro.
- **Probar** - intentar hacer algo.

# Guía docente

## Día 1 - Explora Ideas Para Proyectos (45 min)

### Ejemplos de proyectos

Objetivo: esta parte del proceso es una exploración. Los estudiantes verán un escenario lleno de ejemplos de proyectos que pueden reutilizar para aprender. Esto no sólo les dará una idea de lo que es posible, sino que también los ayudará a ver las limitaciones de la herramienta.

Permite que los estudiantes tengan un día para jugar y reutilizar los proyectos que se encuentran en **Ejemplos de Proyectos Curso Rápido**. Pídeles que usen sus diarios (o cuadernos) para llevar un registro de sus pensamientos e ideas a medida que avanzan.

Esta actividad debe realizarse con las mismas parejas/grupos que trabajarán juntos en los proyectos durante las próximas lecciones.

Asegúrate de que tu clase entienda que durante las próximas semanas trabajarán en sus propios proyectos, por lo que deben prestar mucha atención a cómo se escribieron esos programas, así como a los conceptos que utilizan.

## Día 2 - El Proceso de Diseño (45 min)

### Definir y Preparar

Meta: Los/as estudiantes presentarán un proyecto y planificarán su estrategia para programar ese proyecto en un solo día. Deben tener un bosquejo del proyecto y una descripción para cuando termine el día.

Preparación de los/as estudiantes para el proceso:

La responsabilidad más importante que usted tiene al iniciar este segmento, es ayudar a su clase a entender el alcance de este proyecto. Los/as estudiantes deben tener claro cuáles son las diversas expectativas para las próximas semanas, de modo que puedan prepararse para su revisión y presentaciones de manera apropiada.

Para ayudar a su clase a manejar este proyecto de múltiples etapas, se les debe entregar el primer día de la planificación tanto la **Diseño del Proyecto Final - Ficha** como la **Proyecto Final de Fundamentos de Ciencias de la Computación - Rúbrica**. Los/as estudiantes podrán entonces seguir la rúbrica de cada paso del camino para predecir cuál será la calificación de su proyecto al final.

La Guía de Diseño del Proyecto Final proporcionará un lugar para que los/as estudiantes capturen pensamientos y procesos relevantes a medida que avanzan, para que estén más preparados/as para sus revisiones y presentaciones al final.

Como profesor/a, debe descargar una copia de los documentos y decidir qué elementos son importantes para usted. Asegúrese de editar o eliminar cualquier cosa que no tenga la intención de atraer la atención del/la estudiante.

Definir y Preparar:

Ahora que la clase tiene su Guía de Diseño del Proyecto Final en la mano, deben comenzar a llenar las preguntas del **Día 1**.

Es probable que los/as estudiantes tengan que acudir a sus notas al revisar los proyectos de ejemplo, especialmente si no tienen acceso a los niveles de proyectos de Artista o Play Lab en línea mientras planifican.

Los/as estudiantes deben concentrarse en definir y planificar su proyecto durante el Día 1, y no pasar a la construcción hasta que sus ideas hayan sido escritas y/o dibujadas.

#### 💡 Sugerencia para la lección:

Deje 5 minutos al final del día para que los/as estudiantes intercambien sus Guías de Diseño Proyecto Final para ver el trabajo de los otros/as. Esto ayudará a asegurarse de que no se ha omitido o pasado por alto algo.

Si los/as estudiantes se quedan estancados/as, ayúdelos/as con sus ideas haciendo preguntas y recordando ejemplos, en lugar de ofrecer soluciones.

## Día 3. Construye Tu Proyecto (45 min)

### Intentar

Objetivo: Los/as estudiantes usarán este día para construir una versión inicial de su proyecto.

Equipados con su Guía de Diseño de Proyecto Final, los/as estudiantes deben dirigirse a los computadores para comenzar a dar vida a sus proyectos.

Este proceso se completará con muchas pruebas y errores. Es probable que los proyectos se conviertan en versiones truncadas del alcance original (si no se modifican por completo). Recuerde a los/as estudiantes que este tipo de compromiso es común en el diseño de software, pero deben asegurarse de documentar las razones de los cambios en su producto.

No deje que la clase se olvide de llenar sus Guías de Diseño del Proyecto Final a medida que avanzan. Podría ser útil sugerir que las parejas/grupos tomen un descanso y trabajen con la guía para comenzar a discutir estas preguntas a mitad de su tiempo en el laboratorio. Alternativamente, el navegador puede mantener los ojos abiertos para obtener respuestas pertinentes mientras el conductor programa.

Asegúrese de que cada miembro del equipo tenga su propia Guía de Diseño del Proyecto Final, ya que hay preguntas sobre los pensamientos y comportamientos individuales de cada estudiante que necesitan ser capturados a lo largo del camino.

## Día 4 (Recomendado para 5º Grado) - Revisa Tu Proyecto (45 min)

### Reflexionar e Intentar nuevamente

Objetivo: Los/as estudiantes trabajarán con otro grupo para dar y recibir retroalimentación en función de hacer el proyecto del otro más fuerte.

#### Reflexionar

Para las reflexiones, haga que cada grupo se junte con otro para probar su proyecto. Después de unos 10 minutos, pida a los grupos que discutan las preguntas en la Guía de Diseño del Proyecto Final.

Anime a los/as estudiantes a hacer las preguntas en la guía y a escribir los comentarios de los equipos que lo han revisado para que puedan volver a consultarlo más tarde. Esta debe tomar aproximadamente 15 minutos más.

#### Intentar nuevamente

Con sus nuevas reflexiones en la mano, los/as estudiantes pueden volver a sus computadores para hacer ediciones. Con sólo 10 minutos para terminar, es probable que tengan que seleccionar sólo la retroalimentación más importante para incorporar.

#### 💡 Sugerencia para la lección:

Los/as profesores/as deben evitar asignar la parte final del trabajo del proyecto como tarea a menos que estén seguros de que los/as estudiantes viven muy cerca uno del otro y tienen acceso a Internet en casa.

## Día 5 & 6 - Presenta Tu Proyecto (45 min cada uno)

### Presentaciones

Objetivo: Los/as estudiantes crearán y presentarán sus proyectos de una manera aprobada (escrita, oral o usando multimedia).

Crear:

Idealmente, tendrá tiempo de clase disponible para que sus estudiantes trabajen en sus presentaciones. Esto les permitirá incorporar ricos componentes multimedia, como **Google Slides**. Para otras ideas de presentación, visite **72 formas creativas de que tus estudiantes muestren lo que saben - Sitio web**.

#### 💡 Lesson Tip:

Si está buscando una sección de esta serie para asignar como tarea, ¡ésta es! Los proyectos no tienen que ser presentados en forma electrónica, así que esta es una gran opción fuera de línea.

Anime a los/as estudiantes a incluir toda la información de la Sección J de la Guía de Diseño del Proyecto Final en su presentación, así como dos o más preguntas de la Sección K.

Presente:

Los/as estudiantes deben mostrar sus aplicaciones primero, luego pueden discutir las preguntas que cubrieron en sus presentaciones.

Puede ser muy útil que los/as estudiantes se anoten en un orden específico para dar sus presentaciones, de modo que puedan disfrutar de las demostraciones de sus compañeros/as de clase, sin preocuparse de si serán llamados/as a continuación.

## Actividad de Extensión

Si sus estudiantes ya se sienten cómodos con los conceptos de codificación, motíuelos a que intenten crear sus proyectos en otra plataforma, como **Scratch** o **Alice**.

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

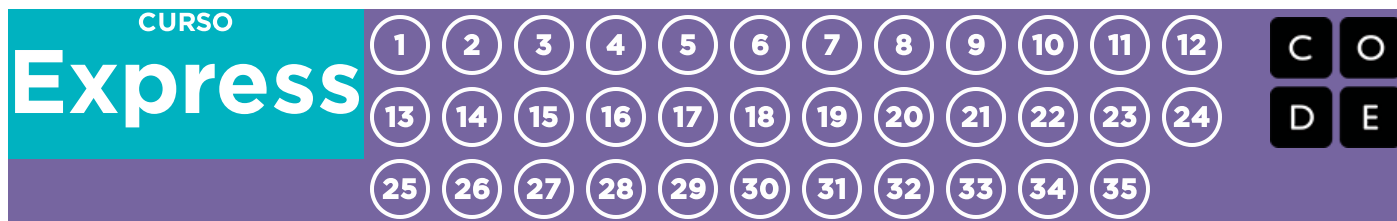
► **AP** - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



# Lección 32: El Proceso de Diseño

Proyecto

## Descripción

En el transcurso de cinco lecciones, los/as estudiantes irán construyendo un proyecto de diseño propio utilizando Play Lab o Artista como entorno de programación. En esta parte del proyecto, los/as estudiantes aprenderán sobre el proceso de diseño y cómo implementarlo en sus propios proyectos, [here](#).

## Propósito

Los/as estudiantes pueden estar listos/as para ir directamente a la construcción de sus proyectos, pero esta lección ayudará a darle forma a sus ideas y transformarlas en planes. Esta estructura mantendrá a los/as soñadores/as con los pies en la tierra e iluminará el camino para aquellos/as que se sienten en la oscuridad.

## Agenda

**Día 2 - El Proceso de Diseño (45 min)**

Definir y Preparar

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Convertir ideas en objetivos y planes razonables.
- Reconocer los posibles obstáculos, como limitaciones de tiempo o errores.

# Guía docente

## Día 2 - El Proceso de Diseño (45 min)

### Definir y Preparar

Los/as estudiantes inventarán un proyecto y planificarán su estrategia para programar ese proyecto en un día. Ellos/as deberían tener un bosquejo y una descripción al final del día.

Preparando a los/as estudiantes para el Proceso:

La responsabilidad más importante que usted tiene en dar el inicio a este segmento es ayudar a su clase a comprender el alcance de este proyecto. Ellos/as deberían tener claro las expectativas de las semanas que siguen para que así puedan prepararse para sus presentaciones de manera adecuada.

Para ayudar a su clase a manejar esta labor de varios pasos, se les debería entregar tanto la **Diseño del Proyecto Final - Ficha** como la **Proyecto Final de Fundamentos de Ciencias de la Computación - Rúbrica** el primer día de planificación. Los/as estudiantes podrán entonces seguir cada paso de la rúbrica para predecir la nota que su proyecto tendrá al final.

La **Diseño del Proyecto Final - Ficha** proporcionará un lugar para que los/as estudiantes capturen los pensamientos y los procesos relevantes mientras trabajan, de esa manera estarán mejor preparados/as para sus presentaciones al final.

Como docente, usted debería decidir qué elementos de estos documentos son importantes para usted y asegurarse de editar o remover cualquier cosa que no necesite del foco de los/as estudiantes.

[Recomendaciones para la lección][3]

Reserve 5 minutos al final del día para que los/as estudiantes intercambien su **Diseño del Proyecto Final - Ficha** para que vean los trabajos de otros. Esto ayudará a asegurarse que nada ha sido omitido o pasado por alto.

[/][3]

Definir y Preparar:

Ahora que la clase tiene la **Diseño del Proyecto Final - Ficha** en sus manos, deberían comenzar a completar las preguntas del **Día 1**.

Los/as estudiantes posiblemente necesitarán revisar sus notas de cuando jugaron con los proyectos de ejemplo, especialmente si no tienen acceso a los niveles de proyectos en Artista o Play Lab mientras planifican.

Ellos/as deberán enfocarse en definir y planificar sus proyectos durante el Día 1 y no pasar a la construcción hasta que sus ideas estén escritas y/o las hayan dibujado.

Si los/as estudiantes se estancan, ayúdelos/as haciendo preguntas y recordando ejemplos, más que ofrecerles la solución.

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

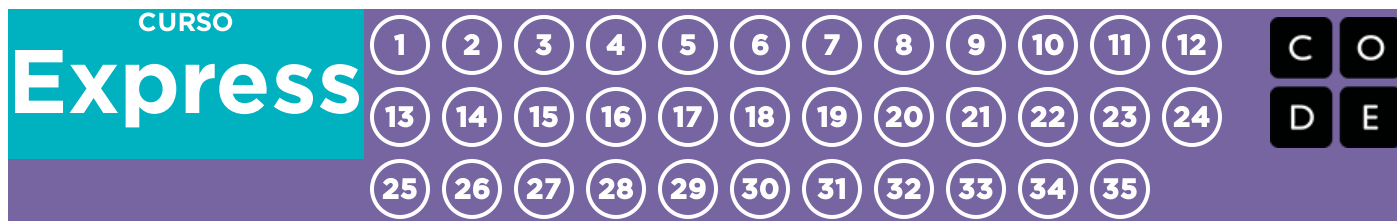
► **AP** - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



# Lección 33: Construye Tu Proyecto

Proyecto

## Descripción

En el transcurso de cinco lecciones, los/as estudiantes irán construyendo un proyecto de diseño propio utilizando Play Lab o Artista como entorno de programación. Ahora los/as estudiantes tendrán su propio espacio para crear su proyecto con Artista o Play Lab. Esta será la parte más larga del proyecto. La guía de la lección que entrega una reseña de las cinco etapas del proceso, se puede encontrar al principio del proceso del proyecto, [here](#).

## Propósito

Esta lección proporciona a los/as estudiantes suficiente tiempo para construir y revisar sus proyectos. El ensayo y error inevitablemente considerado en esta lección, enseñará sobre la solución de problemas y la perseverancia.

## Agenda

**Día 3. Construye Tu Proyecto (45 min)**

Intentar

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Usar el diseño planificado como modelo para la creación.
- Superar los obstáculos, como limitaciones de tiempo o errores.



# Guía docente

## Día 3. Construye Tu Proyecto (45 min)

### Intentar

Los/as estudiantes usarán este día para construir una versión inicial de su proyecto.

Equipados/as con su **Diseño del Proyecto Final - Ficha**, deberían dirigirse a los computadores para comenzar a dar vida a sus proyectos.

Este proceso se completará con suficiente uso del ensayo y error. Es posible que los proyectos sean versiones truncadas del original (si es que no totalmente cambiado). Recuérdeles que este tipo de cambios es común en el diseño de software, pero deben asegurarse de documentar las razones para los cambios en su producto.

No deje que la clase se olvide de llenar sus **Diseño del Proyecto Final - Ficha** a medida que avanzan. Podría ser útil sugerir que las parejas/grupos tomen un descanso en la hoja de trabajo para comenzar a discutir estas preguntas a mitad de su tiempo de laboratorio. Alternativamente, el navegante puede mantener los ojos abiertos para obtener respuestas pertinentes mientras el conductor codifica.

Asegúrese de que cada miembro del equipo tenga su propia Hoja de Trabajo de Diseño del Proyecto Final, ya que hay preguntas sobre los pensamientos y comportamientos individuales de cada estudiante que necesitan ser capturados a lo largo del camino.

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

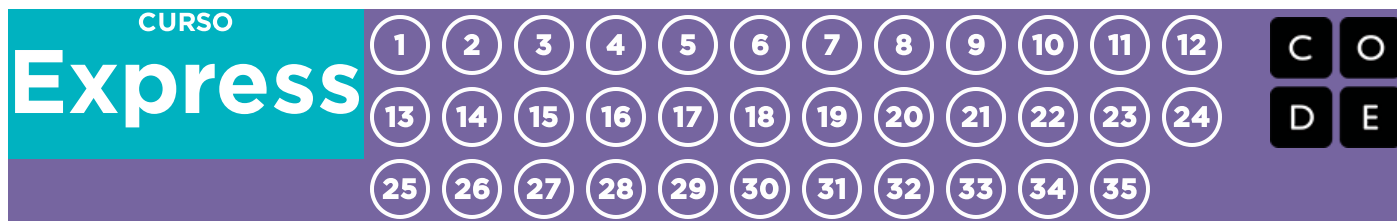
► **AP** - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



# Lección 34: Revisa Tu Proyecto

Proyecto

## Descripción

En el transcurso de cinco lecciones, los/as estudiantes estarán construyendo un proyecto de su propio diseño usando ya sea Play Lab o Artista como su entorno de programación. Ahora que los proyectos están contruidos, los/as estudiantes tienen la oportunidad de obtener retroalimentación de sus compañeros/as y revisar sus proyectos. La reseña de las cinco etapas del proceso se puede encontrar al principio del proceso del proyecto, [here](#).

## Propósito

Esta lección ayuda a los/as estudiantes a dar un paso atrás y ver su proyecto desde una nueva perspectiva. Aquí, los/as estudiantes podrán decidir si han alcanzado sus objetivos. Si no lo han hecho, esta lección les da tiempo y espacio para completar el proyecto.

## Agenda

**Día 4 - Revisa Tu Proyecto (45 min)**

**Reflexionar e Intentar nuevamente**

**Ver en Code Studio**

## Objetivos

Los estudiantes podrán:

- Determinar si los criterios establecidos en una rúbrica se han cumplido en el proyecto actual.
- Redactar e implementar planes para resolver cualquier problema en el código.

# Guía docente

## Día 4 - Revisa Tu Proyecto (45 min)

### Reflexionar e Intentar nuevamente

Objetivo: Los/as estudiantes trabajarán con otro grupo para dar y recibir retroalimentación en función de hacer el proyecto del otro más fuerte.

#### Reflexionar

Para las reflexiones, haga que cada grupo se junte con otro para probar su proyecto. Después de unos 10 minutos, pida a los grupos que discutan las preguntas en la Guía de Diseño del Proyecto Final.

Anime a los/as estudiantes a hacer las preguntas en la guía y a escribir los comentarios de los equipos que lo han revisado para que puedan volver a consultarlo más tarde. Esta debe tomar aproximadamente 15 minutos más.

#### Intentar nuevamente

Con sus nuevas reflexiones en la mano, los/as estudiantes pueden volver a sus computadores para hacer ediciones. Con sólo 10 minutos para terminar, es probable que tengan que seleccionar sólo la retroalimentación más importante para incorporar.

#### 💡 Sugerencia para la lección:

Los/as profesores/as deben evitar asignar la parte final del trabajo del proyecto como tarea a menos que estén seguros de que los/as estudiantes viven muy cerca uno del otro **y** tienen acceso a Internet en casa.

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

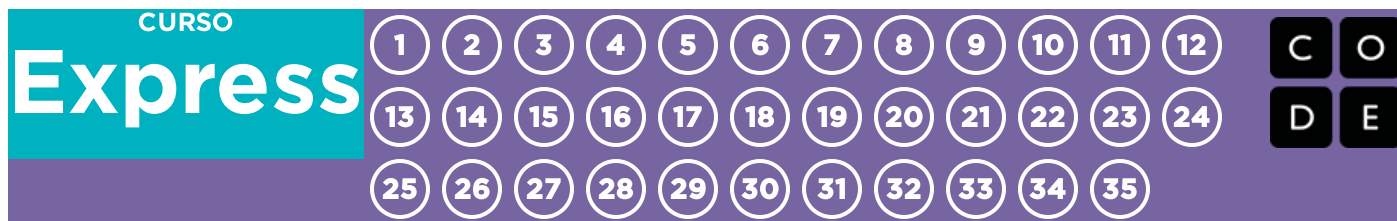
► AP - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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



# Lección 35: Presenta Tu Proyecto

Proyecto

## Descripción

En el transcurso de cinco lecciones, los/as estudiantes irán construyendo un proyecto de diseño propio utilizando Play Lab o Artista como entorno de programación. Finalmente, los/as estudiantes podrán presentar su trabajo terminado a sus compañeros/as o compartirlo con sus seres queridos, con un link especial. La guía de lecciones que resume las cinco etapas del proceso se encuentra al principio del proceso del proyecto, [here](#).

## Propósito

En este punto, los/as estudiantes han trabajado muy duro en sus proyectos, por lo que esta lección pretende ofrecer un espacio para que compartan sus proyectos. Esta lección construirá una comunidad de apoyo donde los/as estudiantes construirán su propia confianza y se sentirán conectados con sus esforzados/as compañeros/as.

## Agenda

**Día 5 & 6 - Presenta Tu Proyecto (45 min cada uno)**

**Presentaciones**

**Ver en Code Studio**

## Objetivos

Los estudiantes podrán:

- Indicar claramente dónde se cumple cada punto de criterio de la rúbrica en el código del proyecto finalizado.
- Articular el proceso de diseño y cómo ayudó a dar forma al proyecto finalizado.

# Guía docente

## Día 5 & 6 - Presenta Tu Proyecto (45 min cada uno)

### Presentaciones

Los/as estudiantes crearán y presentarán sus proyectos de una manera aprobada (escrita, oral o usando multimedia).

#### Crear:

Idealmente, tendrá tiempo de clase disponible para que sus estudiantes trabajen en sus presentaciones. Esto les permitirá incorporar ricos componentes multimedia, como **Google Slides**. Para otras ideas de presentación, visite **72 formas creativas de que tus estudiantes muestren lo que saben - Sitio web**.

Anime a los/as estudiantes a incluir toda la información de la Sección J de la Guía de Diseño del Proyecto Final en su presentación, así como dos o más preguntas de la Sección K.

#### Presente:

Los/as estudiantes deben mostrar sus aplicaciones primero, luego pueden discutir las preguntas que cubrieron en sus presentaciones.

Puede ser muy útil que los/as estudiantes se anoten en un orden específico para dar sus presentaciones, de modo que puedan disfrutar de las demostraciones de sus compañeros/as de clase, sin preocuparse de si serán llamados/as a continuación.

#### 💡 Lesson Tip:

Si está buscando una sección de esta serie para asignar como tarea, ¡ésta es! Los proyectos no tienen que ser presentados en forma electrónica, así que esta es una gran opción fuera de línea. Otras formas de presentar sus proyectos en línea o sin conexión, pueden ser:

- Reporte
- Publicación en un blog
- Online
- Frente a la clase con un afiche

## Normas de aprendizaje

Estándares para ciencias de la computación CSTA K-12 (2017)

- ▶ **AP** - Algoritmos y programación



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

Español (Latinoamérica) ▼

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