

# Curso E

Creado para los/as estudiantes de 5° básico, este curso comienza con una breve revisión de los conceptos enseñados previamente en los cursos C y D. Esta introducción pretende inspirar a los/as principiantes y recordar a los/as expertos/as las maravillas de la Ciencia de la Computación. Los/as estudiantes practicarán codificando algoritmos, loops, condicionales y eventos, antes que conozcan las funciones. Al final del curso, los/as estudiantes tendrán la oportunidad de crear un proyecto final que podrán compartir orgullosos/as con sus pares y otros seres queridos.

Links para el/la docente: [Teacher Videos Playlist](#)

## Lección 1: Programación: Mis Amigos Robóticos

Algorithms | Debugging | Unplugged

¡Convierte a tus amigos en robots y diles qué hacer!

## Lección 2: Secuencias en Laberinto

Sequencing | Debugging | Maze

¡En esta lección, aprenderás a escribir tus propios programas!

## Lección 3: Construyendo los Cimientos

Unplugged | Persistence | Frustration

Construye una estructura que pueda sostener un libro. Puede que te sientas frustrado/a.

¡Recuerda ser perseverante!

## Lección 4: Debugging en Scrat

Debugging | Ice Age

Encuentra problemas en los desafíos de La Era del Hielo y practica tus habilidades de debugging.

## Lección 5: Programación en Artista

Program | Programming | Artist

Crear imágenes hermosas programando al Artista.

## Lección 6: Mis Amigos Robóticos Loopy

Unplugged | Loop | Repeat

¡Convierte a tus amigos en robots y diles qué hacer usando loops!

## Lección 7: Loops en Artista

Loop | Artist

En esta lección, ¡los loops facilitan el hacer imágenes incluso más geniales con artista!

## Lección 8: Loops Anidados

Nested Loops | Loops | Bee | Maze

Loops dentro de loops dentro de loops. ¿Qué significa? Esta lección te enseñará qué sucede cuando colocas un loop dentro de otro loop.

## Lección 9: Loops Anidados con Frozen

Loop | Nested Loop | Frozen

Anna y Elsa tienen excelentes habilidades de patinaje para hielo, pero necesitan tu ayuda para crear patrones en el hielo. Utiliza loops anidados para crear algo super COOL.

## Lección 10: Algoritmos: Carrera de Dados Sin Conexión

Unplugged | Dice Race | Algorithms

En esta lección de algoritmos, aprende a describir un juego desde el punto de vista del computador.

## Lección 11: Introducción a los Desafíos Online

Sequencing | Debugging | Loop | Ice Age | Maze | Artist

Esta lección te dará la práctica en las habilidades que necesitarás para este curso.

## Lección 12: Condicionales en Granjero

Conditionals | Farmer

Podrás decirle al computador qué hacer bajo ciertas condiciones en esta serie divertida y desafiante.

## Lección 13: Ciudadanía Digital

Common Sense Education | Personal Information | Private Information | Identity Theft

Internet es entretenido y emocionante, pero es importante mantener la seguridad también. Esta lección te enseñará la diferencia entre la información que es segura de compartir y la que es privada.

## Lección 14: Construir un Juego de Star Wars

Star Wars | Event

Siente la fuerza mientras construyes tu propio juego de Star Wars en esta lección.

## Lección 15: Funciones: Composición de Canciones Sin Conexión

Unplugged | Function

Incluso las estrellas de rock necesitan habilidades de programación. Esta lección te enseñará sobre funciones usando letras de canciones.

## Lección 16: Funciones en Artista

Function | Artist

¡Haz dibujos complejos de manera más fácil usando funciones!

## Lección 17: Funciones en Abeja

Function | Bee

No escribas muchos códigos para juntar todo el néctar y la miel. ¡En lugar de eso usa funciones!

## Lección 18: Funciones en Cosechadora

Function | Harvester

¡Las funciones te ahorrarán mucho trabajo mientras ayudas a la granjera con su cosecha!

## Lección 19: Determinar el Concepto

Bee

¡No estamos revelando ningún secreto! Esta lección puede utilizar cualquiera de las habilidades que has aprendido hasta ahora.

## Lección 20: Construir un Juego de Play Lab

Event | Play Lab

Practica haciendo juegos para compartir con tus amigos/as y familia.

## Lección 21: Explorar Ideas de Proyectos

Project | Define | Prepare | Try | Reflect

Aquí hay diferentes juegos y diseños. ¡Juega con cada uno de ellos para obtener ideas para tus propios proyectos!

## Lección 22: El Proceso de Diseño

Project

Los proyectos así de grandes toman tiempo y mucha planificación. Aquí, aprenderás acerca del proceso de diseño que usarás para construir tu propia creación.

## Lección 23: Construye Tu Proyecto

Project

¡Prepara esas manos para programar mucho! Es hora de empezar a construir tu proyecto.

## Lección 24: Presenta Tu Proyecto

Project

¡Prepárate para presumir! Es tiempo de presentar tu proyecto finalizado a tus compañeros/as.

## Lección 25: 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 26: Más Allá de la Programación: Crowdsourcing

Unplugged | Crowdsourcing

Esta lección te enseñará sobre crowdsourcing, el proceso de construir un proyecto con un equipo.



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: Mis Amigos Robóticos

Algoritmos | Depurar | Sin conexión

## Descripción

Utilizando un conjunto especial de comandos offline, los/as estudiantes diseñarán algoritmos para dar instrucciones a un "robot" para apilar vasos en diferentes patrones. Los/as estudiantes tomarán turnos participando como el robot, respondiendo sólo al algoritmo definido por sus pares. Este segmento enseña a los/as estudiantes la conexión entre símbolos y acciones, la diferencia entre un algoritmo y un programa, y la valiosa habilidad de debugging.

## Propósito

Esta lección sin conexión une a la clase como equipo con la tarea simple que completar: hacer que un "robot" apile unos vasos de una forma específica. Los/as estudiantes trabajarán para reconocer acciones del mundo real como instrucciones potenciales en códigos. También se practicará el arte de seguir instrucciones precisas, a medida que los/as estudiantes trabajan para traducir algoritmos a códigos, usando los símbolos proporcionados. Si surgen problemas en el código, los/as estudiantes deberían trabajar juntos para reconocer los errores y construir soluciones. Esta actividad sienta las bases para la programación que los/as estudiantes harán a lo largo del curso mientras aprenden la importancia de definir un algoritmo claramente comunicado.

## Agenda

### Actividad previa (5 min)

Hablar con Robots

### Actividad (45 min)

Introducción y Modelación

Opciones de diferenciación:

Programando Sus Robots

### Actividad de Cierre (10 min)

Escribir en el Diario

## Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Replantear una secuencia de pasos como un programa codificado
- Identificar y corregir bugs o errores en instrucciones en secuencia

## Preparación

☐ Vea el **Mis Amigos Robóticos - Video para el docente**.

☐ (Opcional) Imprima un **Programación: Mis Amigos Robóticos - Clave de símbolo** por grupo o cada 4 estudiantes.

Alternativamente, encuentre un lugar donde poner esta información donde los/as estudiantes puedan usar como referencia durante la lección.

☐ Prepare una pila de 10 vasos desechables por grupo de 4 estudiantes, O

☐ (Opcional) imprima y corte **resource paper-trapezoid-template not found** para cada grupo si su clase no va a usar vasos.

☐ Imprima un set de **Apilando Ideas - Manipuladores** por 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 estudiantes

- **Programación: Mis Amigos Robóticos -** Plantilla de trapecios en papel  
[Hacer una copia](#)
- **Apilando Ideas -** Manipuladores
- **Programación: Mis Amigos Robóticos -** Video sin conexión ([descargar](#))

- **Programación: Mis Amigos Robóticos** - Clave de símbolo

## Vocabulario

- **Algoritmo** - una lista de tareas para finalizar una tarea.
- **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.
- **Repetir** - hacer algo de nuevo.

# Guía docente

## Actividad previa (5 min)

### Hablar con Robots



**Discutir** Comience preguntando a la clase si alguien ha oído sobre la robótica. ¿Alguien ha visto alguna vez un robot o tocado uno? ¿Un robot "escucha" realmente lo que tu hablas? ¿"Entiende" realmente lo que estás diciendo?

**Decir** Los robots sólo pueden hacer lo que han sido instruidos o programados para hacer. Para realizar una tarea, un robot necesita tener una lista de instrucciones (a veces llamada algoritmo) que pueda leer. Hoy, vamos a aprender lo que se necesita para que eso suceda.

#### Discussion Goal

El objetivo de esta discusión rápida es llamar la atención sobre el hecho de que, si bien los robots parecen comportarse como personas, en realidad sólo responden a su programación. Los/as estudiantes probablemente se referirán a los robots de las películas y la televisión que se comportan más como humanos. Motívelos a considerar robots que han visto u oído en la vida real, como los Roombas, o incluso asistentes digitales como Amazon Alexa.

## Actividad (45 min)

### Introducción y Modelación



**Mostrar:** Muestre el **Programación: Mis Amigos Robóticos - Clave de símbolo** o escriba las acciones permitidas en la pizarra - asegúrese que estén en un lugar dónde puedan verlos durante toda la actividad. Explique a la clase que estas serán las únicas seis acciones que usarán durante el ejercicio. Para esta tarea, instruirán a su amigo/a "robot" para que construya una pila de vasos específica usando sólo los comandos enlistados en la clave.

**Modelar:** Con el propósito de explicar cómo deben funcionar las instrucciones, modele para la clase cómo crear y seguir un algoritmo para replicar un patrón simple. Ubique una pila de vasos simple en frente suyo para comenzar.

**Mostrar:** Sostenga el patrón que planea modelar. Un patrón simple de tres vasos es un buen lugar para comenzar.



**Pick Up Cup**



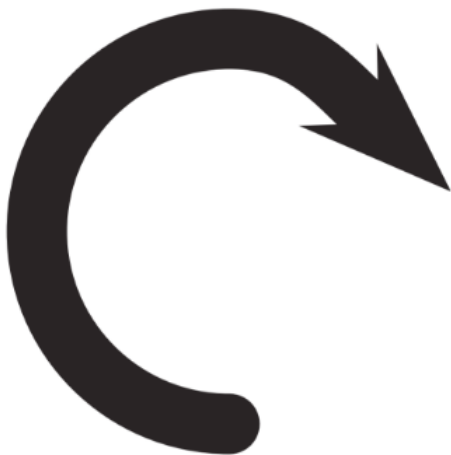
**Put Down Cup**



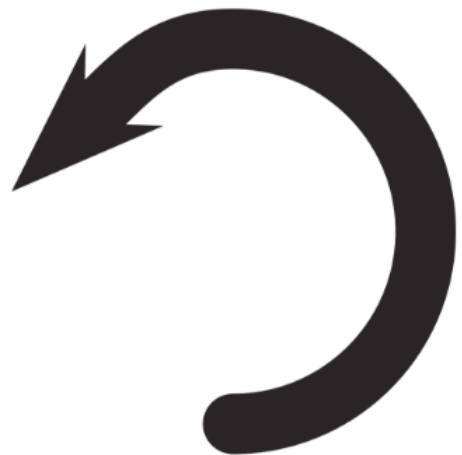
**Step Forward**



**Step Backward**



**Turn Cup Right 90°**



**Turn Cup Left 90°**



## Opciones de diferenciación:

**Simplificar:** ¿Todo esto se siente un poco complicado para su clase? He aquí un par de consejos para simplificar el proceso:

- Cargue muchos pasos en hacia arriba y hacia abajo
  - Arriba significa que el vaso sube automáticamente tan alto como sea necesario
  - Abajo significa que baja automáticamente hasta que aterrice sobre algo
  - La mano regresa automáticamente a la pila de vasos después de posicionar un vaso

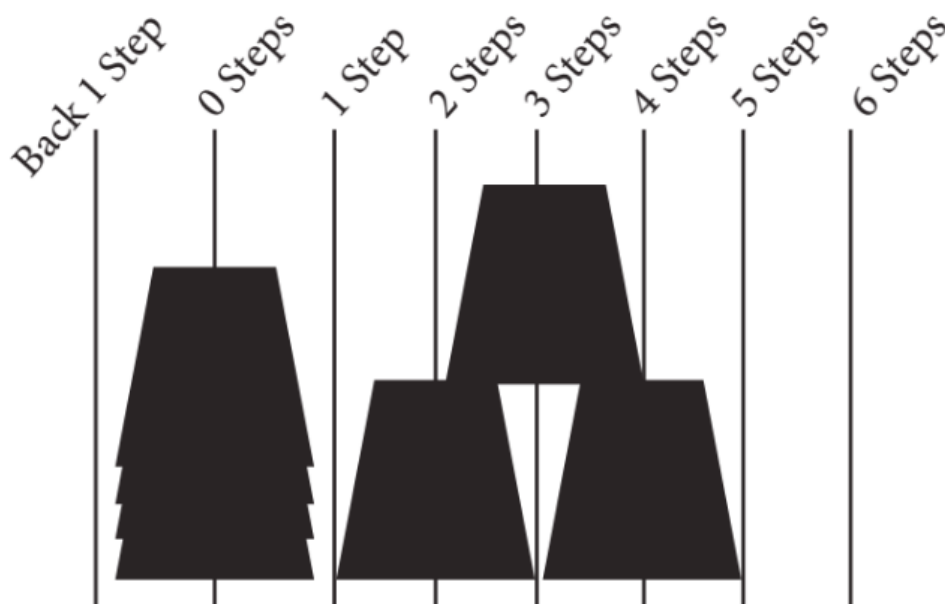
**Intensificar:** ¿Son sus estudiantes más avanzados/as? ¿Quiere que esta lección esté relacionada más cercanamente a los desafíos online? Aquí hay algunas modificaciones que puede hacer:

- Una flecha corresponde a un movimiento
  - Cuando se remueve un vaso de la pila, vuelve a la mesa antes de moverlo.
  - Los/as estudiantes necesitan usar múltiples flechas "hacia arriba" para levantar el vaso varias veces
  - Los/as estudiantes necesitan usar varias flechas "hacia abajo" para bajar los vasos varios niveles
  - Los/as estudiantes necesitan usar las flechas "hacia atrás" para devolver el vaso a la pila
  - **Montaje:** Tenga pilas de vasos o recortes de trapezoides de papel disponible para los grupos.



**Sugerir:** Pregunte a la clase cuál sería la primera instrucción, usando *sólo las seis instrucciones permitidas*. El primer movimiento debería ser "recoger vaso". Si los/as estudiantes sugieren algo más de la lista, realice esa acción y permítales ver su error. Si sugieren algo que no está en la lista, muestre una reacción clara de mal funcionamiento e infórmeles que el comando no fue comprendido.

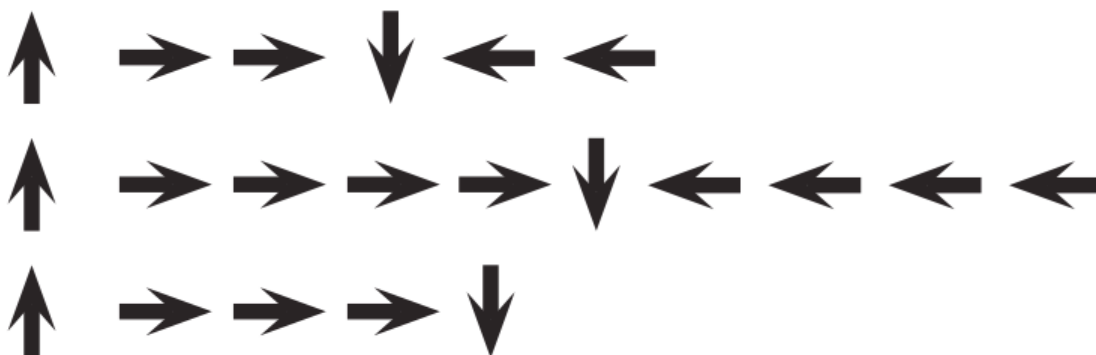
Con un vaso en la mano, pida a la clase que continúe dándole instrucciones hasta que el primer vaso esté ubicado. Este es un buen momento para clarificar que un "paso al frente" y un "paso atrás" implican moverse el ancho de medio vaso. Vea las imágenes siguientes como referencia.



Continúe pidiendo instrucciones a la clase hasta que hayan completado el diseño completo.

Una vez que su pila esté completa, señale que ellos/as acaban de darle una serie de instrucciones para completar una tarea. Eso es un algoritmo. Los algoritmos son geniales para compartir ideas, pero deletrearlos palabra por palabra puede tomar mucho tiempo. ¡Para eso son los símbolos! Cuando codifican un algoritmo en símbolos que un robot (o computador) comprende, eso se llama programación.

Pida a la clase que lo/a ayude a escribir el "programa" para ese primer movimiento, y luego el resto de los movimientos necesarios para completar el patrón. Dependiendo de la confianza de sus estudiantes, podría cambiar alternativamente entre actuar como el "robot" y escribir el código, o puede incentivarlos para que escriban todo el programa antes que lo implemente. Una solución posible luce algo así:



**Voluntario/a:** Una vez que la clase haya completado el programa modelo, pida a uno/a de los/as estudiantes que salga adelante y actúe como el "robot" para asegurar que el programa realmente funciona. Anímelos/as a decir las instrucciones en voz alta a medida que "ejecutan" el código.

**Agrupar:** Coloque a los/as estudiantes en grupos de 4. Cada grupo debe dividirse en dos parejas - cada pareja desarrollará su propio programa para ser "dirigido" por la otra pareja.

**Distribuir:** Dé a cada grupo una pila de vasos o recortes de papel..

**Mostrar:** Muestre **Apilando Ideas - Manipuladores** a la clase, o distribuya copias individuales para que los grupos las usen. Haga que cada pareja (no grupo) escoja qué idea les gustaría hicieran sus robots. Trate de que opten por una idea más fácil la primera vez, y luego pídale que escojan un diseño más complejo más adelante. Anime a las parejas a mantener su elección en secreto de la otra mitad de su grupo.

**Discutir:** Dé a cada pareja tiempo para discutir cómo se debe construir la pila, usando sólo los símbolos proporcionados. Asegúrese de que cada grupo anote el "programa" en algún lugar, para que el "robot" lo lea más tarde.



**Hacer:** Una vez que ambas parejas del grupo han decidido sus algoritmos, pueden turnarse para ser "robots", siguiendo las instrucciones que cada pareja escribió. Anime a los/as estudiantes a observar de cerca a su "robot" para asegurarse de que están siguiendo las instrucciones. Si un/a estudiante ve un bug y levanta la mano, pídale al robot que termine las instrucciones lo mejor que pueda. Después, pida a los/as estudiantes que discutan el problema potencial y que propongan una solución. Continúe repitiendo hasta que la pila se construya correctamente.

#### 💡 Teaching Tip

**Reforzando las reglas:** Mientras el robot esté trabajando en la pila asegúrese que la clase sepa:

- A los/as programadores/as no se les está permitido hablar cuando el robot está trabajando. Esto incluye dar respuestas o señalar cuando el robot ha hecho algo mal.
- Los/as programadores/as deben levantar sus manos si ven un bug.

**Circular:** Busque grupos que estén intentando tomar atajos añadiendo cosas (como números) a su código. Elógielos/as por su ingenio, pero recuérdelos que para este ejercicio, los robots no entienden *nada más* que los símbolos proporcionados. Si desea, puede sugerirles que guarden su brillante solución para la próxima vez que jueguen a este juego, ¡ya que podrían tener la oportunidad de usarla pronto!.

**Repetir:** Dependiendo del tiempo disponible, mezcle las parejas y deles la oportunidad de hacer un patrón diferente. Cada vez que los grupos repitan el proceso, animelos a elegir un patrón más desafiante.



**Discutir:** Después que todos hayan tenido la oportunidad de ser el robot, reúna a la clase para discutir su experiencia. En particular, discutan como clase:

- ¿Cuál fue la parte más difícil de encontrar las instrucciones?
- ¿Alguien encontró un bug en sus instrucciones, mientras su robot las estaba siguiendo?
  - ¿Cuál era el bug?
  - ¿Por qué crees que no te diste cuenta cuando escribiste el programa?
- Cuando eras el robot ¿cuál fue la parte más difícil de seguir las instrucciones que te dieron?

#### 💬 Discussion Goal

**Tener Sentido:** El objetivo de esta discusión es dar a los/as estudiantes espacio para que entiendan su experiencia como robots y como programadores. Las preguntas son intencionalmente amplias, pero están diseñadas para que los estudiantes piensen en los desafíos de escribir un programa claro y las limitaciones de un robot o computadora para interpretar sus instrucciones.

## 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?
- Dibuja una pila de vasos que el robot hizo hoy.
- ¡Dibuja una pila de vasos que te gustaría que hiciera un robot algún día!

## 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: Secuencias en Laberinto

Secuencia | Depurar | Laberinto

## Descripción

En este grupo de desafíos, los/as estudiantes comenzarán con una introducción (o revisión dependiendo de la experiencia de su clase) al espacio de trabajo online de Code.org. Habrá videos señalando la funcionalidad básica del espacio de trabajo incluyendo los botones "Ejecutar", "Reiniciar" y "Paso". También se discute en estos videos: Arrastrar los bloques Blockly, borrar bloques Blockly y conectar bloques Blockly. Luego, los/as estudiantes practicarán sus habilidades de secuenciación y debugging en laberinto.

## Propósito

Reconocemos que cada sala de clases 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.

## Agenda

### Actividad previa (10 min)

Introducción  
Vocabulario

### Actividades Puente - Programación (10 min)

Actividad Sin Conexión Usando Bloques de Papel  
Previsualización de los Desafíos Online como Clase

### Actividad Principal (30 min)

Course E Online Puzzles 2018 - Sitio web

### Actividad de Cierre (10 min)

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

- ☐ Recorra **Curso E Desafíos Online - Sitio web** para encontrar cualquier problema potencial para su clase.
- ☐ 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

- **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

## 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 (10 min)

### 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.

## Actividades Puente - Programación (10 min)

Esta actividad ayudará a traer los conceptos sin conexión de “Mis Amigos Robóticos” al mundo online al que están entrando los/as estudiantes. Escoja una de las siguientes actividades para realizar con su clase:

### Actividad Sin Conexión Usando Bloques de Papel

De forma similar a “Mis Amigos Robóticos”, ponga a sus estudiantes en parejas. Reparta múltiples bloques “llenar 1” y “mover” de **Bloques Blockly Sin Conexión (Cursos 2-5) - Manipuladores** para cada pareja. Pida a cada pareja de estudiantes dibujar un diseño en una hoja cuadriculada, de cuatro por cuatro, de la **Programación con Papel Cuadriculado - Ficha**. Luego, haga que los/as estudiantes trabajen juntos para escribir el programa que se necesita para dibujar este diseño usando los bloques Blockly de papel. Los/as estudiantes necesitarán escribir “hacia arriba”, “hacia abajo”, “a la derecha” o “a la izquierda” en el bloque “mover\_\_”. Asegúrese que los/as estudiantes sepan que el programa va de arriba hacia abajo y que ¡los bloques deben tocarse!

### Previsualización de los Desafíos Online como Clase

Seleccione un desafío desde **Course E Online Puzzles 2018 - Sitio web**. Recomendamos el Desafío 6 para esta actividad. Divida a los/as estudiantes en grupos de 3 o 4. Pídales que “programen” a Red, el Angry Bird para que llegue al cerdo usando las flechas de “Mis Amigos Robóticos”.



Una vez que todos los grupos tienen una respuesta, discutan el camino como clase.

## Actividad Principal (30 min)

### Course E Online Puzzles 2018 - Sitio web

Los/as profesores/as son de vital importancia en la educación en las Ciencias de la Computación y en apoyar un ambiente de aula vibrante y colaborativa. Durante las actividades online, el rol del/la profesor/a es de estimular y apoyar. Las lecciones online están preparadas para ser centradas en el/a estudiante, así que los/as profesores/as deberían evitar involucrarse cuando los/as estudiantes se estanquen. Algunas ideas de cómo hacer esto son:

- Use **Programación en Parejas - Video para el estudiante** cada vez que sea posible durante la actividad.
- Anime a los/as estudiantes con desafíos/preguntas para comenzar preguntándole a su compañero/a.
- Las preguntas sin responder pueden ser escaladas a un grupo cercano, que podría ya tener la respuesta.
- Recuerde a los/as estudiantes usar el proceso de debugging antes que usted se acerque.
- Pida a los/as estudiantes que describan el problema que están viendo. ¿Qué se supone que debe hacer? ¿Qué hace? ¿Qué te dice eso?
- Recuérdele a los/as estudiantes frustrados/as que la frustración es un paso en el camino del aprendizaje, y que la perseverancia tendrá sus frutos.
- Si un/a estudiante aún está estancado/a después de todo esto, hágale preguntas claves para hacer que los/as estudiantes identifiquen el error por ellos/as mismos/as.

#### 🔔 Sugerencia para el/la Profesor/a:

Enseñe a los/as estudiantes la forma correcta de ayudar a los/as compañeros/as:

- No te sientes en el asiento de tu compañero/a.
- No uses el teclado del/la compañero/a.
- No toques el mouse del/la compañero/a.[1]
- Asegúrate que el/a compañero/a pueda describirte la solución, antes de que te retires.

## 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?
- Enumera algunos de los bugs que encontraste en tus programas hoy.
- ¿Cuál era su desafío favorito para completar? Dibuja a tu personaje favorito completando un 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 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.
- Revisar el plan y la estructura hasta cumplir el desafío.

## Preparación

- ▢Vea [teacher-video-practicing-persistence][0].
- ▢Vea el [lesson-in-action-foundation][1].
- ▢Imprima [csf-building-a-foundation-activity][2].
- ▢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 [think-spot-journal][3] a cada estudiante.

## Enlaces

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

Para los docentes

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

## 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

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)

Course E Online Puzzles 2018 - 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 **Course E Online Puzzles 2018** - **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 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

- **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)

### Course E Online Puzzles 2018 - 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: Programación en Artista

Programa | Programación | Artista

## Descripción

En esta lección, los/as estudiantes tomarán el control del Artista para completar los dibujos en pantalla. Esta etapa del Artista permitirá a los/as estudiantes crear imágenes de creciente complejidad usando nuevos bloques como moverse hacia adelante por 100 píxeles y girar a la derecha por 90 grados.

## Propósito

Partiendo de la experiencia previa de los/as estudiantes con la secuenciación, esta lección trabajará para inspirar más creatividad con la programación. El propósito de esta lección es consolidar el conocimiento de la secuenciación mediante la introducción de nuevos bloques y objetivos. En este caso, los/as estudiantes aprenden más sobre los píxeles y ángulos usando los nuevos bloques, mientras practican sus habilidades de secuenciación. Además, los/as estudiantes podrán visualizar nuevas metas, tales como programar al Artista para dibujar un cuadrado.

## Agenda

### Actividad previa (10 min)

Introducción

### Actividad Principal (30 min)

Course E Online Puzzles 2018 - Sitio web

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

Escribir en el Diario

### Aprendizaje Extendido

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Crear un programa para completar una imagen mediante pasos secuenciales.
- Dividir formas complejas en partes simples.

## Preparación

- ☐ Recorra **Curso E Desafíos Online - Sitio web** para encontrar cualquier área de problemas potencial para su clase.
- ☐ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones**.
- ☐ (Opcional) Tener transportadores para que su clase visualice los ángulos que deben usar para completar los desafíos.
- ☐ Imprima una **Giros & Ángulos - Guía para el estudiante** para cada estudiante.

## 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

- **Diario Think Spot** - Diario de reflexiones [Hacer una copia ▾](#)
- **Introducción a Artista** - Video para el estudiante
- **Giros & Ángulos** - Video para el estudiante
- **Giros & Ángulos** - Guía para el estudiante [Hacer una copia ▾](#)

## Vocabulario

- **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 (10 min)

### Introducción

Muestre a los/as estudiantes uno o ambos de los siguientes videos como introducción a los ángulos:

**Introducción a Artista - Video para el estudiante** (1,5 minutos de duración)

**Giros & Ángulos - Video para el estudiante** (2 minutos de duración)

Use el **Giros & Ángulos - Guía para el estudiante** para mostrar los ángulos interiores y exteriores de los alumnos para diferentes formas. Este documento se puede usar como una entrega o puede elegir imprimirlo como un póster para que los estudiantes consulten.

Pedir:

Discuta las formas cuadradas y triangulares del documento.

- ¿Cómo codificarías una computadora para dibujar esa forma?
- ¿En qué orden deben estar las instrucciones?

Dígalas a los/as estudiantes que en estos desafíos moverán a un personaje que deja una línea donde quiera que vaya. Los estudiantes escribirán un código que hace que el personaje dibuje varias formas, incluyendo un cuadrado.

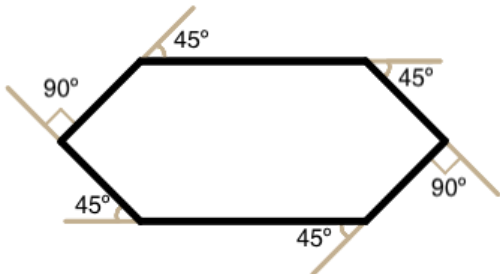
## Actividad Principal (30 min)

### Course E Online Puzzles 2018 - Sitio web

En este grupo de desafíos, el artista ya no estará limitado a ángulos de  $90^\circ$ . Tener transportadores disponibles para la clase puede ayudar a los/as estudiantes a visualizar de mejor manera los ángulos que necesitan. De lo contrario, la lección proporciona imágenes de los ángulos a medida que el/a estudiante selecciona cuál usar. (Por favor considerar: Las elecciones de los ángulos está limitada a dos dentro del menú desplegable, reduciendo el número de opciones con las que tienen que trabajar los/as estudiantes).

Antes de enviar a los/as estudiantes a trabajar en los computadores a resolver los desafíos, podría ser beneficioso hacer una breve presentación sobre cómo usar las herramientas en este nivel. Recomendamos el desafío 5 como un buen ejercicio para mostrar cómo usar el transportador online.

El octavo desafío pide a los/as estudiantes dibujar un polígono de seis lados. Esto podría ser desafiante para algunos/as de ellos/a. Recomendamos permitirles que lo intenten unas pocas veces, preguntar a un compañero/a, luego pedirle ayuda al/la profesor/a. A continuación, una imagen que podría ser útil para los/as estudiantes.



## Actividad de Cierre (10 - 15 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?
- ¿Cuáles son los ángulos interiores que forman un cuadrado? ¿Qué hay de un triángulo?
- Dibuja una forma simple en tu papel e imagina el código utilizado para dibujarla. ¿Puedes escribir ese código junto a la forma?

## Aprendizaje Extendido

Use las siguientes actividades para mejorar el aprendizaje de sus estudiantes. Pueden ser utilizados como actividades fuera de la clase o en otra oportunidad.

### La fotocopidora

- Entregue a los/as estudiantes dos hojas de papel.
- En una de las hojas, dibuje una imagen sencilla, usando sólo líneas rectas.
- En la segunda hoja, dibuje las instrucciones para recrear la imagen usando comandos para moverse en línea recta y girar en distintos ángulos.
- Intercambie las hojas de instrucciones e intente recrear la imagen usando sólo las instrucciones entregadas.

## 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: Mis Amigos Robóticos Loopy

Sin conexión | Bucle | Repetir

## Descripción

Basándose en la actividad inicial "Mis amigos Robóticos", los/as estudiantes aprenden a usar loops cuando programan sus robots para construir estructuras más grandes de manera más eficiente.

## Propósito

Esta lección sirve como una re introducción a los loops, usando el ahora familiar set de instrucciones de programación para el/a "robot". Los/as estudiantes desarrollarán habilidades de pensamiento critico buscando patrones de repetición en los movimientos de los/as compañeros/as y determinando cómo simplificar esos patrones repetidos usando loops.

## Agenda

### Actividad previa (10 min)

Revisión Mis Amigos Robóticos

### Actividad (30 min)

Introducción y Modelación

### Haciendo Loops con tus Robots

### Actividad de Cierre (5 min)

Conversación Rápida / Escribir en el diario

### Actividades de Extensión

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Identificar patrones repetidos en el código que podrían reemplazarse con un bucle
- Escribir instrucciones que usen bucles para repetir patrones

## Preparación

- ☐ Asegúrese de que cada estudiante tenga un [think-spot-journal][0]
- ☐ Revise [my-loopy-rf-teacher-vid][1]
- ☐ Prepare una pila de 20 vasos de papel para cada grupo de 4 estudiantes.
- ☐ (Opcional) Imprima un [my-loopy-rf-packet][2] por grupo de 4 estudiantes.

\*\*O\*\*

- ☐ Muestre el [symbol-key][3] al que los/as estudiantes pueden dirigirse durante la clase.
- ☐ Imprima y recorte [paper-trapezoid-template][4] para cada grupo si su clase no va a usar los vasos.
- ☐ Imprima un set de [stacking-cups-ideas][5] por grupo.

## Enlaces

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

Para los estudiantes

- **Paquete de Mis Amigos Robóticos Loopy** - PDF

## Vocabulario

- **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.
- **Repetir** - hacer algo de nuevo.

# Guía docente

## Actividad previa (10 min)

Revisión Mis Amigos Robóticos



**Pick Up Cup**



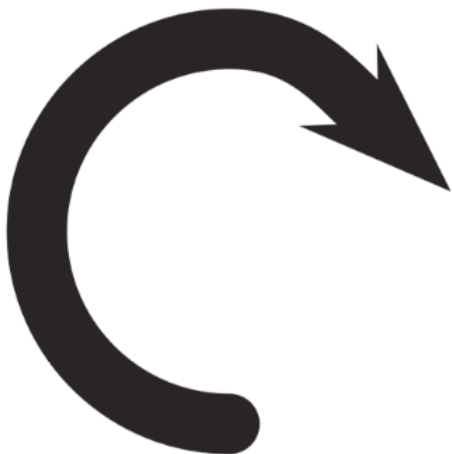
**Put Down Cup**



**Step Forward**



**Step Backward**



**Turn Cup Right 90°**



**Turn Cup Left 90°**



**Objetivo:** Esta revisión refrescará las mentes de los/as estudiantes en relación a lo rápido que los programas de "Mis Amigos Robóticos" se pueden intensificar.

**Demostrar:** Muestre **Programación: Mis Amigos Robóticos - Clave de símbolo** que utilizamos en Mis Amigos Robóticos. Para cada uno de los seis símbolos, pida a los/as estudiantes que muestren cómo se ve que un robot siga esa instrucción.

**Modelar:** Con toda la clase, elija un desafío fácil del Paquete de Pila de Vasos de "Mis Amigos Robóticos" y programen juntos como un recordatorio de las reglas y la terminología.

A continuación, elija un desafío que sea un poco más difícil, pero que también requiera muchos pasos como el anterior.



**Voluntario/a:** Pida a un/a voluntario/a (o un grupo de voluntario/as) que pase adelante para ayudar a programar este desafío en la pizarra. Si les pide que atengan estrictamente a la regla de "no hay más símbolos que los que están en las instrucciones", esto podría tomar un buen rato.

**Mostrar:** Ahora, muestre la siguiente imagen:



¿Cuál es la reacción de la clase?

**Recordar:** Dé a los estudiantes la oportunidad de hacer una lluvia de ideas sobre formas más cortas de transmitir el código que están a punto de crear. (Esta parte puede omitirse si sus alumnos empiezan a decir cosas como: "Avanzar 6 veces". Ya que eso abrirá la discusión sobre cómo mostrar "seis veces" con símbolos.)

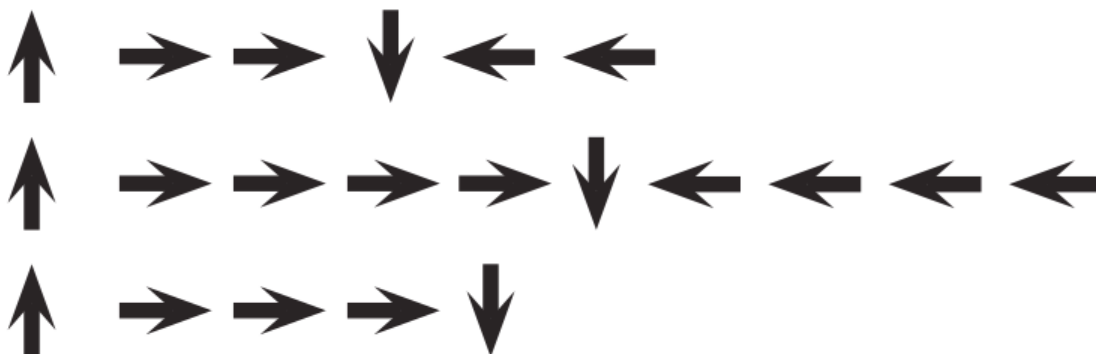
Una vez que los estudiantes hayan llegado a la idea de "repetir" el código, entrégueles el vocabulario que lo rodea. Asegúrese de compartir con ellos que los términos "repetir algo" y "hacer un loop en algo" a menudo se usan indistintamente.

## Actividad (30 min)

### Introducción y Modelación

**Preparación:** Tenga a mano pilas de vasos o trapecios de papel cortado disponibles para los grupos.

**Mostrar:** Tome el programa de algunas de sus pilas de vasos anteriores y muéstrelo a la clase, o utilice el que se encuentra más abajo.

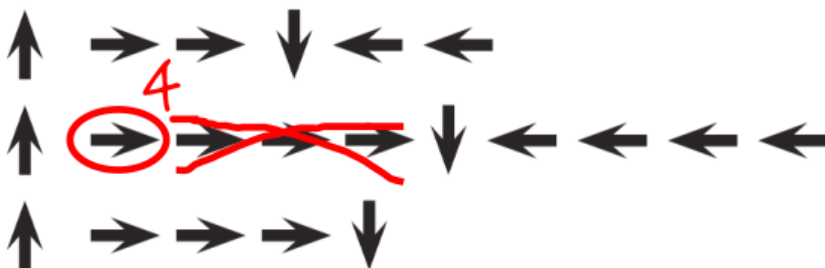


**Pensar:** Pida a los/as estudiantes que piensen en que parte de este programa pueden encontrar un patrón de instrucciones que se repita sin interrupciones (una repetición después de la otra).

**Trabajar en parejas:** Con el compañero/a, comparte uno de los patrones de repetición que encuentre.

**Compartir:** Pida a algunos/as estudiantes que compartan los patrones identificados. Intente nombrar distintos enfoques para agrupar los patrones. Pida a los/as estudiantes que identifiquen cuantas veces se repite cada patrón.

**Modela:** Utilizando uno de los patrones repetidos identificados por la clase, modele como Circular la instrucción o patrón que se repita, escriba el número de loops cerca del círculo, y luego tache el resto de las flechas.



Repita esto hasta que el programa completo se haya acortado, luego vuelva a escribir el programa de forma tal que sus estudiantes puedan ver cuanto más simple terminan siendo las instrucciones.

## Haciendo Loops con tus Robots

**Agrupar:** Agrupe sus estudiantes de a 4. Cada grupo debe luego dividirse en parejas - cada pareja desarrollará su propio programa para que la otra pareja "ejecute".

**Distribuir:** Entregue a cada grupo una pila de vasos o de recortes de papel.

**Mostrar:** Muestre **Apilando Ideas - Manipuladores** a la clase o entregue copias individuales a cada grupo para que utilicen. Haga que cada pareja (no grupo), elija que idea les gustaría que realizara su robot. Motívelos/as a elegir un patrón más complejo esta vez.



**Discutir:** Permita que cada grupo discuta cómo se debe construir la pila, y luego instruya a cada grupo para que traduzca el algoritmo a los símbolos. Asegúrese de que cada grupo anote el programa en algún lugar para que el "robot" lo lea más tarde. Mientras los estudiantes trabajan en sus programas, recuérdelos que deben estar atentos a las oportunidades de reemplazar un patrón de repetición por un loop.

**Hacer:** Cuando los grupos hayan terminado sus instrucciones, haga que cada pareja se turne para "ejecutar" su código con otra pareja. Recuerde a los/as estudiantes que deben estar atentos a los errores de su código, pero que no deben interrumpir al robot hasta que haya terminado de ejecutar el programa.

**Discutir:** Cuando todas las parejas hayan tenido la oportunidad de ejecutar sus programas, pida a algunas que compartan sus soluciones con la clase. Aproveche esta oportunidad para discutir sobre cómo los

### Teaching Tip

**Buscando Loops:** Asegúrese de mantener los ojos abiertos para ver si los estudiantes usan loops. Trate de evitar corregir sus algoritmos generales o entregar la solución, pero siéntase libre de dirigir a los/as estudiantes hacia patrones que podrían acortarse usando un círculo de repetición.

Observe a los/as estudiantes mientras ejecutan el código. ¿Hay algún bug? Utilice las preguntas de debugging para ayudarles a encontrar una solución.

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

grupos llegaron a diferentes soluciones para el mismo desafío. En particular, puede preguntar sobre cada programa:

- ¿Cómo identificaron los loops?
- ¿Hay otras formas de escribir esos loops?
- ¿Cuánto más corto es el programa con loops de lo que sería sin este?
- ¿Es más fácil entender el programa con loops o escrito a mano? ¿Por qué?

## Actividad de Cierre (5 min)

### Conversación Rápida / Escribir en el diario

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

Conversación Rápida:

Aquí hay algunos posibles temas:

- ¿Sientes que los loops hacen la programación más fácil o más difícil?
- Qué otro tipo de cosas repetimos en la vida?
  - Comer - poner comida en la boca, masticar 20 veces
  - Peinarse - cepillarse el pelo 35 veces
  - Rutinas - Despertar, ir al colegio, volver a casa, irse a la cama
- Escribe o dibuja algo en tu diario que te recuerde más adelante lo que son los loops.
  - ¿Qué significa para ti "repetir"?
  - Haz un dibujo de ti repitiendo algo.

## 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 pídale 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 7: Loops 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 Principal (30 mín)

Course E Online Puzzles 2018 - 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

- Recorra el **Curso E Desafíos Online - Sitio web** asociado a este nivel, para encontrar cualquier área potencial de problema para 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**.

## 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

- **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

Los/as estudiantes deberían haber tenido una gran introducción a los loops al llegar a este punto. Basado en lo que usted piensa que podría beneficiar a su clase, recomendamos:

- Crear un nuevo diseño de agrupación con loops tal y como en “Mis Amigos Robóticos Loopy”.
- Previsualizar un desafío de esta lección.

Todas estas opciones harán una revisión de los loops o del artista, ¡lo que ayudará a preparar a su clase para divertirse con los desafíos online!

## Actividad Principal (30 min)

### Course E Online Puzzles 2018 - Sitio web

Algunos/as estudiantes podrían descubrir dónde agregar los loops “repetir” escribiendo el programa sin loops y luego haciendo un círculo en las secciones de repeticiones. Si los/as estudiantes en su clase parecieran beneficiarse con esto, pídeles que tengan papel y lápices al lado de sus computadores. Los/as estudiantes podrían disfrutar también dibujar algunas de las formas y figuras en el papel antes de programarlas online. (Cuando dibujan estampillas, puede ser más fácil simbolizar éstos 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 8: Loops Anidados

Loops Anidados | Bucles | Abeja | Laberinto

## 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)

Course E Online Puzzles 2018 - Sitio web

### Actividad de Cierre (15 min)

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

☐ Recorra **Course E Online Puzzles 2018 - Sitio web** para encontrar cualquier problema potencial 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 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

- **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)

### Course E Online Puzzles 2018 - Sitio web

Recomendamos altamente el **Programación en Parejas - Video para el estudiante** en esta lección. Este puede no ser un tema fácil para la mayoría de los/as estudiantes. Trabajar con un/a compañero/a y discutir soluciones potenciales para los desafíos puede aliviar la preocupación de los/as estudiantes.

También, tenga papel y lápices a mano para que los/as estudiantes escriban su plan antes de codificar. Algunos desafíos tienen un número limitado de ciertos bloques que pueden usar, así que si los/as estudiantes escriben la respuesta larga para encontrar las repeticiones, el papel 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 9: Loops Anidados con Frozen

Bucle | Bucle anidado | Frozen

## 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 min)

Course E Online Puzzles 2018 - Sitio web

### Actividad de Cierre (15 mín)

Escribir en el Diario

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Describir 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 a través de **Course E Online Puzzles 2018 - 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 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

- **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)

### Course E Online Puzzles 2018 - Sitio web

Este conjunto de desafíos se configura como una progresión. Esto significa que cada desafío crea una base para el siguiente. Los/as estudiantes disfrutarán de hacer más y más diseños interesantes haciendo pequeños y simples cambios en el código que ya han escrito.

## 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 10: Algoritmos: Carrera de Dados Sin Conexión

Sin conexión | Carrera de dados | Algoritmos

## Descripción

En esta lección, los/as estudiantes relacionarán el concepto de algoritmos con actividades de la vida real jugando Carrera de Dados. El objetivo es comenzar a construir las habilidades para traducir situaciones de la vida real, los escenarios online y viceversa.

## Propósito

Presentando a la clase un concepto básico como *algoritmos*, los/as estudiantes que se sienten intimidados por los computadores aún pueden construir una base para la comprensión de estos temas. Los algoritmos son esenciales para la ciencia de la computación. En esta lección, los/as estudiantes aprenderán a traducir las instrucciones a un algoritmo que tiene un rol en la programación.

## Agenda

### Actividad previa (15 mín)

Introducción

Vocabulario

### Actividad Principal (30 min)

Algoritmos de la vida real: carrera de dados - Ficha

### Actividad de Cierre (15 mín)

Conversación Rápida: ¿Qué aprendimos?

Escribir en el Diario

### Aprendizaje Extendido

Descúbrelo

### Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Desglosar actividades grandes en una serie de eventos más pequeños.
- Organizar eventos secuenciales en su orden lógico.

## Preparación

- ▢ Vea el [teacher-video-dice-race][0].
- ▢ Imprima una [csf-dice-race-worksheet][1] por grupo.
- ▢ Imprima una [csf-dice-race-assessment][2] por alumno.
- ▢ Entregue a cada estudiante un [think-spot-journal][3].

## Enlaces

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

Para los docentes

- **Algoritmos de la vida real: carrera de dados** - Video sin conexión ([descargar](#))
- **Algoritmos de la vida real: carrera de dados** - Ficha
- **Algoritmos de la vida real: carrera de dados** - Guía de respuestas en fichas
- **Algoritmos de la vida real: carrera de dados** - Evaluación
- **Algoritmos de la vida real: carrera de dados** - Guía de respuestas de la evaluación

Para los estudiantes

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

[Hacer una copia](#) ▼

## Vocabulario

- **Algoritmo** - una lista de tareas para finalizar una tarea.

# Guía docente

## Actividad previa (15 mín)

### Introducción

- Pregunte a los/as estudiantes qué hicieron para prepararse para ir al colegio esta mañana.
  - Escriba sus respuestas en la pizarra.
  - Si es posible, ponga números al lado de sus respuestas para indicar el orden en que suceden.
    - Si los/as estudiantes dan sus respuestas en desorden, haga que lo ayuden a ponerlas en algún tipo de orden lógico.
    - Señale lugares donde el orden importa y lugares donde no importa.
- Presente a los/as estudiantes la idea de que es posible crear algoritmos para las cosas que hacemos todos los días.
  - Deles un par de ejemplos: cómo preparar el desayuno, atarse los zapatos y cepillarse los dientes.
- Los computadores necesitan algoritmos y programas para mostrarles cómo hacer incluso las cosas más simples que nosotros podemos hacer sin pensar en ellas.
  - Puede ser desafiante describir algo que viene de forma natural con suficiente detalle para que lo replique un computador.
- ¡Vamos a intentar hacer esto con una nueva actividad nueva y divertida, como es jugar una Juego de Carrera de Dados!

### Vocabulario

Esta lección tiene una palabra de vocabulario que es importante de revisar:

- **Algoritmo** - Repítanlo conmigo Al — go — rit - mo:

Una lista de pasos para terminar una tarea.

## Actividad Principal (30 min)

### Algoritmos de la vida real: carrera de dados - Ficha

- Puedes usar los algoritmos para ayudar a describir las cosas que la gente hace diariamente. En esta actividad, crearemos algoritmos para describir cómo jugamos el Juego Carrera de Dados.
- La parte más difícil de tener un problema listo para que un computador pueda descifrar como escribir las actividades de la vida real. Vamos a hacer algo de práctica jugando y describiendo el Juego Carrera de Dados.

#### 💡 Lesson Tip

Usted es quien conoce mejor a su clase. Como profesor/a, decida si los/as estudiantes deberían hacer esto en parejas o grupos pequeños.

#### Instrucciones

- Lee las reglas que se encuentran más abajo.
- Juega un par de rondas del Juego Carrera de Dados.
  - Mientras juegas, piensa cómo describirías todo lo que estás haciendo.
  - ¿Cómo luciría desde el punto de vista del computador?

[Lesson Tip][1]

Ayude a los/as estudiantes a ver el juego desde el punto de vista del computador. Si necesitan lanzar los dados, entonces el computador necesita proporcionar dados. Si el/a estudiante necesita jugar 3 turnos, entonces el computador necesita hacer un loop por los pasos varias veces.

[/][2]

### Reglas:

- Ponga a todos los jugadores en puntaje 0.
- Pida al primer jugador que lance los dados.
- Agregue los puntos de ese lanzamiento al puntaje total del/a jugador/a.
- Pida al siguiente jugador que lance.
- Agregue los puntos de ese lanzamiento al puntaje total del/a jugador/a dos.
- Cada jugador debería repetir unas dos veces más.
- Revise el puntaje total de cada jugador, para ver quién tiene más puntos.
- Declare un/a ganador/a.

Juego 1	Turno 1	Turno 2	Turno 3	Total
Jugador 1				
Jugador 2				

¡Encierra en un círculo a el/la ganador/a!

Junte a la clase y pida a cada estudiante que complete la **Algoritmos de la vida real: carrera de dados - Evaluación**. Una vez que los/as estudiantes han finalizado la hoja de trabajo, pídeles compartir sus algoritmos con la clase. Abra una discusión sobre la diferencia entre un algoritmo desde un punto de vista de un/a humano/a y el punto de vista de un computador.

## Actividad de Cierre (15 mín)

### Conversación Rápida: ¿Qué aprendimos?

- ¿Cuántos de ustedes pudieron seguir los algoritmos de sus compañeros/as para jugar el Juego de Carrera de Dados?
- ¿Cuál es la diferencia entre un algoritmo y un programa?
  - Un algoritmo es el pensamiento detrás de lo que se necesita que ocurra, mientras el programa es el grupo de instrucciones reales que hacen que suceda.
  - Un algoritmo debe ser traducido a un programa antes que el computador pueda usarlo.
- ¿Dejó algo fuera el ejercicio?
  - ¿Qué habrías agregado para hacer el algoritmo incluso mejor?
  - ¿Qué pasaría si el algoritmo hubiese sido sólo de un paso: “Jugar la Carrera de Dados”?
    - ¿Habría sido más fácil o más difícil?
    - ¿Qué habría pasado si fuesen cuarenta pasos?
- ¿Cuál fue tu parte favorita de la actividad?

### 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 es un algoritmo?
- ¿Cuáles son algunos algoritmos que usas en tu vida diaria?

## Aprendizaje Extendido



Use estas actividades para profundizar el aprendizaje de los/as estudiantes. Pueden usarse como una actividad fuera del aula o cualquier otro enriquecimiento.

## Descúbrelo

- Separe al curso en equipos.
- Pida a cada equipo que invente varios pasos en los que puedan pensar para completar una tarea.
- Junte nuevamente a los equipos en un gran grupo y pida a un equipo que comparta sus pasos, sin decir qué actividad era la que habían escogido.
- Permita que el resto de la clase intente adivinar para qué actividad es el algoritmo.

## 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 11: Introducción a los Desafíos Online

Secuencia | Depurar | Bucle | Era de Hielo | 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

### Actividad Puente (15 mín)

Previsualización de los desafíos en línea

### Actividad Principal (30 min)

Curso E Desafíos Online - 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

- ☐ Recorra los desafíos **Curso E Desafíos Online - Sitio web** para encontrar cualquier problema potencial para su clase.
- ☐ 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

- **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

- **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

Hoy se dedicará en gran medida a lograr que los/as estudiantes se introduzcan en Code.org o que proporcionen una revisión. Brevemente, pregunte su clase sobre su experiencia con Code.org. Si hay una amplia gama de estudiantes, junte a los/as más experimentados/as con estudiantes no experimentados/as durante sus primeras o segundas lecciones. Si no hay una amplia gama de experiencia, proponga una discusión sobre qué es Code.org y por qué se está enseñando.

## Actividad Puente (15 mín)

Esta actividad ayudará a llevar los conceptos sin conexión de "Carrera de dados" al mundo en línea en el que los estudiantes avanzan. Le recomendamos realizar la siguiente actividad con la clase:

### Previsualización de los desafíos en línea

Escoja un desafío de **Curso E Desafíos Online - Sitio web**. Recomendamos el desafío 9. Dígale a la clase que Scrat de La Era del Hielo llegar a la bellota. Para que Scrat llegue a su objetivo, la clase necesita hacer un programa. Asegúrese de mostrar a la clase cómo usar las pistas y el botón avanzar al completar este desafío.

## Actividad Principal (30 min)

### Curso E Desafíos Online - Sitio web

Los/as 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 online están diseñadas para centrarse en los estudiantes, por lo que los/as 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/as estudiantes con preguntas/desafíos que deben comenzar por preguntar a sus compañeros/as.
- Preguntas sin responder pueden ser escaladas a un grupo cercano, quienes pueden ya tener la respuesta.
- Recuerde a los/as estudiantes que pueden utilizar el proceso de debugging 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/as estudiantes que se sientan frustrados/as, que la frustración es un paso en el camino del aprendizaje, y que la perseverancia valdrá la pena.
- Si un/a estudiante sigue estancado/a después de todo esto, haga preguntas que lo/la lleven a descubrir el error por sí mismo/a.

#### 💡 Recomendación para el/la Profesor/a

Muestre a los/as estudiantes la forma **correcta** de ayudar a sus compañeros/as:

- No sentarse en la silla de el/la compañero/a
- No usar el teclado de el/la compañero/a
- No tocar el mouse de el/la compañero/a
- Asegúrese de que el/la compañero/a le pueda describir la solución antes de que usted se retire.

## 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 12: Condicionales en Granjero

Condicionales | Granjera

## Descripción

Esta lección introduce a los/as estudiantes a los loops "while" y "si/ si no". Los *Loops While* son loops que continúan repitiendo comandos mientras sean verdaderos. Los Loops While se usan cuando el/la programador/a no conoce exactamente el número de veces que un comando necesita repetirse, pero si sabe que la condición debe ser verdadera para permitir que el loop continúe repitiéndose. Las afirmaciones "Si/si no" ofrecen flexibilidad en la programación, ejecutando secciones enteras de códigos sólo si algo es verdadero, de lo contrario ejecuta otra cosa.

## Propósito

Una comprensión básica de los condicionales es un pre-requisito necesario para el Curso E. Creamos esta introducción para hacer una revisión para los/as estudiantes que ya están familiarizados/as con los condicionales y les permite practicar a aquellos/as que están recién aprendiendo. Si encuentra que la comprensión de los condicionales varía demasiado en su clase, le recomendamos un trabajo estratégico en parejas cuando trabajen en esta lección online.

## Agenda

### Actividad previa (15 mín)

Introducción

### Actividad Principal (30 mín)

Course E Online Puzzles 2018 - 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

- Recorra los desafíos en **Curso E Desafíos Online - Sitio web** para encontrar cualquier problema potencial 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 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

- 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.
- **Bucles mientras** - un bucle que se sigue repitiendo mientras una condición es verdadera.

# 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)

### Course E Online Puzzles 2018 - Sitio web

Los patrones en estos desafíos puede que no sean obvios para todos los/as estudiantes. Recomendamos que recorra estos niveles antes para comprender mejor cualquier área de problemas para su clase. También, ver y usar las técnicas del **Programación en Parejas - Video para el estudiante** puede ser útil para la clase.

## 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?
- ¿Qué es un condicional? ¿Por qué programarías un condicional?
- Da un ejemplo del uso de un condicional durante tu día (ejemplo: Si tengo hambre, Como algo; mientras camino por la calle, mantengo la vista en los autos)

## Aprendizaje Extendido

### Mientras Jugamos

¡Reúna a la clase en el patio o en un gimnasio con una pelota! Esto puede ser hecho en círculo o como equipos en una cancha.



**\*Reglas:**

- *Mientras* la pelota esté en juego, debemos estar preparados/as para golpearla
- *Si* la pelota es lanzada hacia ti, debes mantenerla en el aire
- *Si* golpeas la pelota una vez, no puedes golpearla de nuevo (sólo un toque por persona, no toques dobles)
- *Si* la pelota sale de los límites, todos deben caerse al suelo dramáticamente. El/la último/a tiene que ir a buscar la pelota.

Al final de la primera ronda, pregunte a los/as estudiantes si pueden identificar los condicionales en el juego.  
¿Pueden crear otros que quisieran en el juego?

## 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: Ciudadanía Digital

Common Sense Education | Información personal | Información privada | Robo de identidad

## Descripción

Desarrollada por Common Sense Education, esta lección trata de diferenciar entre la información que es segura de compartir online y la información que no lo es. A medida que los/as estudiantes visitan sitios que requieren información acerca de sus identidades, aprenden a adoptar un proceso de investigación crítico que los/as empodera para protegerse a ellos/as mismos/as y a sus familias de ladrones/as de identidades. En esta lección, los /as estudiantes aprenden a pensar de forma crítica acerca de la información de usuario/a que algunos sitios web requieren o solicitan. Aprenden la diferencia entre información privada e información personal, así como a diferenciar lo que es seguro o inseguro de compartir online.

## Propósito

Common Sense Education ha creado esta lección para enseñar a los/as niños/as la importancia de la seguridad en Internet. Por medio de la discusión acerca de la diferencia entre información privada y personal, los/as estudiantes podrán reconocer qué información debería ser compartida y cual no. Los/as estudiantes también aprenderán qué señales deberían buscar para determinar si un sitio web es seguro o no.

## Agenda

### Actividad previa (5 min)

Introducción

### Actividad Principal (35 min)

Acceder

Privado y Personal

¿Qué es Seguro Compartir Online?

### Actividad de Cierre (15 min)

Conversación Rápida: ¿Qué aprendimos hoy?

Escribir en el Diario

### Evaluación (10 min)

Información Privada y Personal

### Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Conocer los beneficios y riesgos de compartir información en línea.
- Comprender qué tipo de información puede ponerlos ante el riesgo de robo de identidad y otras estafas.

## Preparación

- ▣ Haga una copia para cada estudiante de la *Guía del estudiante Protégete Tú Mismo* (Séptima página de la guía de preparación docente).
- ▣ Haga una copia para cada estudiante de la *Guía del estudiante Todo Sobre Mí* (Sexta página de la guía de preparación docente).
- ▣ Imprima una evaluación para cada estudiante (Octava página de la guía de preparación docente).
- ▣ Revise previamente sitios web como **Neopets**, **Nickelodeon**, y **BookAdventure** y prepárelos para mostrarlos a la clase.
- ▣ Revise **Ciudadanía Digital de Fundamentos de Ciencias de la Computación - Lista de recursos** para mayor contenido sobre seguridad online.

## Enlaces

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

Para los docentes

- **Common Sense Education - Información Privada y Personal** - Guía de preparación para el docente
- **Common Sense Education** - Sitio web
- **Ciudadanía Digital de Fundamentos de Ciencias de la Computación** - Lista de recursos

Para los estudiantes

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

Hacer una copia ▼

## Vocabulario

- **Robo de identidad** - cuando un ladrón roba información privada de alguien para pretender ser esa persona.
- **Información personal** - información que no se puede utilizar para identificarte.
- **Información privada** - información que se puede utilizar para identificarte.
- **Registrarse (en línea)** - ingresar tu información con el fin de registrarte y obtener acceso a un sitio web.

# Guía docente

## Actividad previa (5 min)

### Introducción

Pregunte:

- ¿Qué tipos de información piensas que está bien compartir públicamente online como en un perfil online que otros verán?
  - Intereses y actividades favoritas.
  - Opiniones acerca de una película.
  - Nombre de pila.
- ¿Cuáles son algunos ejemplos de sitios webs donde debes registrarte para participar?
  - Sitios de Redes Sociales.
  - Sitios para compartir videos.
  - Sitios de discusión para jóvenes.
  - Sitios para preguntarle a un/a experto/a.
  - Sitios de juegos.

**Escriba** los nombres de los sitios web en la pizarra. Explique que es importante saber que compartir ciertos tipos de información como usuario, puede poner su privacidad y la de su familia en riesgo.

## Actividad Principal (35 min)

[[0]



Como alternativa offline, imprima y copie las páginas del sitio web que piden información para registrarse e iniciar sesión. Distribúyalo a los/as estudiantes.

[/][1]

### Acceder

**Proyecte** para la clase o pídale que vayan al sitio **Neopets**, **Nickelodeon**, or **BookAdventure**. *iNo les pida que se inscriban en estos sitios!*

**Discuta** con los/as estudiantes los tipos de información que cada sitio solicita antes de que el/a usuario/a pueda participar.

Pregunte:

- ¿Qué información se pide? ¿Por qué crees que se pide?
- Nombre de pila, nombre de usuario, clave, pista de clave, género, la región dónde vives, permiso de los padres, etc. Se pide esta información porque ayuda a distinguir una persona de la otra. O quizás el sitio web lleva un registro de quien lo usa.
- ¿Qué información es opcional? ¿Por qué crees que es opcional?
- Email de los padres, cumpleaños, región, ciudad, país, género, etc. Es probable que esta información sea opcional porque el sitio web no la requiere para pago o para distinguir gente. O quizás el sitio web quiere llevar registro de este tipo de información.
- ¿Por qué crees que los sitios webs piden este tipo de información?
- Quieren que la gente pague para usar el sitio, quieren enviar mensajes a la gente que están ingresando, o quieren intentar vender cosas a esa gente.

**Señale** que no es necesario completar todos los campos en los sitios webs si no son requeridos. Los campos requeridos están usualmente marcados con un asterisco (\*) o se destacan en rojo.

## Privado y Personal

**Explique** a los/as estudiantes que es seguro compartir algunos tipos de información en internet y algunos no lo son. Sin embargo, la información que se considera segura no debería ser compartida en privado con gente que no se conoce offline.

Defina:

- **Información Personal:** Información que no puede ser usada para identificarte.
- **Información Privada:** Información acerca de ti, pero que no puede ser usada para identificarte.

Enfatice que usualmente es seguro compartir online la información personal. Usualmente no es seguro compartir la información privada online, lo que quiere decir que los/as estudiantes deberían tener permiso de uno de los padres o tutor legal, antes de compartir este tipo de información.

### 💡 Sugerencia para el/la Profesor/a

Si quisiera una distinción más clara entre información “personal” y “privada” en estas definiciones, puede usar otras frases como “información amigable” o “información compartible” para definir mejor la línea que los/as estudiantes debieran reconocer. Decidimos mantener “personal” y “privada” para ser fieles a la planificación Common Sense Education.

**Comparta** los siguientes ejemplos de información que es segura compartir y la que no lo es:

SEGURO - Información Personal	NO SEGURO - Información Privada
<ul style="list-style-type: none"><li>- Tu comida favorita</li><li>- Tu opinión (aunque debe ser hecha con respeto)</li><li>- Nombre de pila (con permiso)</li></ul>	<ul style="list-style-type: none"><li>- Nombre de la madre</li><li>- Número de carnet</li><li>- Tu fecha de nacimiento</li><li>- Información de la tarjeta de crédito de los padres</li><li>- Número de teléfono</li></ul>

Pregunte:

- ¿Por qué alguien querría robar la identidad de otra persona en internet?
  - Para robar dinero.
  - Hacer algo malo.
  - Esconder su identidad real.

Defina:

- **Robo de identidad:** cuando un/a ladrón/a roba la información privada de alguien con el propósito de pretender ser esa persona.

**Explique** que el/a ladrón/a de identidad usa la información privada para pretender ser la persona cuya identidad, él o ella, ha robado. Una vez que el/a ladrón/a ha tomado la identidad de alguien, él o ella puede usar el nombre de esa persona para obtener una licencia de conducir o comprar cosas, ¡incluso si esa persona cuya identidad robaron no tenga la edad suficiente para hacer esas cosas! A menudo, no es hasta mucho más tarde que la gente se da cuenta que su identidad ha sido robada. Los/as ladrones/as de identidad también podrían obtener una tarjeta de crédito a nombre de otras personas y acumular grandes cuentas que luego no pagan. Informe a los/as estudiantes que los/as ladrones/as de identidad a menudo se enfocan en niños/as y adolescentes porque tienen una historia limpia y sus padres difícilmente estarán conscientes de que alguien está robando la identidad de sus hijos/as.

Enfatice la diferencia entre información privada (que puede ser utilizada para robar tu identidad) e información personal (que no puede ser usada para robar tu identidad). Invite a los/as estudiantes a responder las siguientes preguntas (escriba sus respuestas en la pizarra).

Pregunte: • ¿Qué tipo de información privada podría un ladrón/a de identidad usar para averiguar y robar tu identidad? - Nombre de pila, dirección, números de teléfono, claves, números de tarjetas de crédito, número de carnet, nombre de la madre.

- ¿Qué tipo de información personal podrías compartir tú mismo sin mostrar tu identidad?

- Edad, género, cuántos hermanos tienes, música favorita, comida favorita, que mascotas tienes, el nombre de tu mascota, tu opinión acerca de algo.

**Explique** a los/as estudiantes que en internet, la gente con la que interactúan pueden ser sus amigos del barrio o extraños que viven al otro lado del mundo. Debido a que es difícil saber las intenciones de la gente que nunca has conocido antes, es mejor ser cautelosos/as cuando compartan información. No deberían darle su información privada a extraños en el mundo real, y deben ser igual de cautelosos/as cuando estén online.

**Recuerde** a los/as estudiantes lo importante que es que cada vez que comparten información online se detengan a pensar: “¿Estoy dando información que debería mantener en privado?” Señale que a veces puede ser seguro entregar información privada. Por ejemplo, un sitio web puede pedirles su fecha de nacimiento o dirección email. Pero los/as estudiantes siempre deberían preguntarle a uno de los padres o al tutor antes de entregar información privada.

**Distribuya** la *Guía del Estudiante Protégete tú Mismo/a* y pida a los/as estudiantes que completen la actividad. Revise las respuestas con la clase.

## ¿Qué es Seguro Compartir Online?

**Distribuya** la *Guía Todo acerca de Mí*. Pida a los/as estudiantes que anoten toda la información personal que les gustaría compartir en un perfil público en una comunidad online. Enfátice que, aunque es seguro compartir la información personal online, está bien decidir no compartirla. Recuérdele a los/as estudiantes que todo lo de la lista debería ser seguro de compartir; nada de esa información debería ser información que pueda poner su identidad en riesgo.

**Motive** a los/as estudiantes a compartir sus listas con la clase.

**Pregunte:**

- ¿Hay algo en las listas que pudiera ser usado por un/a ladrón de identidad? ¿Por qué?
  - Guíe a los/as estudiantes a explicar sus respuestas y anímelos a usar las palabras del vocabulario.

## Actividad de Cierre (15 mín)

### Conversación Rápida: ¿Qué aprendimos hoy?

Puede usar estas preguntas para evaluar la comprensión de sus estudiantes acerca de los objetivos de la lección. Puede que quiera pedirle a los/as estudiantes reflexionar escribiendo una de las preguntas, usando su diario o en un blog/wiki online.

**Pregunte:**

- ¿Qué es un robo de identidad?
  - Usar la información privada de alguien para fingir ser esa persona.
- ¿Cómo se diferencian información personal e información privada?
  - Información privada, como el número de carnet, no es seguro de compartir. Debería mantenerse en privado para que los/as ladrones de identidad no la puedan usar. La información personal, como tu comida favorita, no puede ser usada por los/as ladrones de identidad y es seguro compartirla. Aunque es seguro compartir la información personal online, puedes escoger no compartirla, y eso está bien.
- ¿Cuál sería una buena regla para los/as niños/as acerca de entregar información privada?
  - No deberían compartirla online sin el permiso de un/a profesor/a, uno de los padres o tutor/a.

### 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?

- Da un ejemplo de información personal e información privada.
- ¿Qué sitio web usas frecuentemente? ¿Cómo sabes si es un sitio web seguro de usar?

## Evaluación (10 min)

### Información Privada y Personal

Entregue la evaluación a los/as estudiantes. Deles el tiempo necesario para completarla. Si queda algo de tiempo, revise las respuestas con ellos/as.

## 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 14: Construir un Juego de 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 Principal (30 min)

Course E Online Puzzles 2018 - 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

- ☐ Recorra los desafíos en **Curso E Desafíos Online - Sitio web** y encuentre cualquier área potencial de problema.
- ☐ 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**.

## 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

- **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 Principal (30 min)

### Course E Online Puzzles 2018 - Sitio web

Es posible que los/as estudiantes estén muy emocionados de crear su propio juego Star Wars al final de este grupo de desafíos. Si hay tiempo, pídales que planeen lo que quieren que haga el juego. La planificación y preparación los/as ayudará a reconocer de mejor forma los conceptos claves que esta lección está tratando de enseñar. Anímelos/as a compartir y remezclar los juegos de todos al final de la lección.

#### Teacher Tip

Recuérdelo a los/as estudiantes compartir sus trabajos solamente con sus amigos/as cercanos/as o con la familia. Para más información vea o muestre a la clase el video “Detenerse y Pensar” Online.

**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 15: Funciones: Composición de Canciones Sin Conexión

Sin conexión | Función

## Descripción

Una de las estructuras más relevantes en el mundo de la ciencia de la computación es la función. Las funciones (a veces llamadas procedimientos) son mini programas que pueden usarse una y otra vez al interior de programas más grandes. Esta lección ayudará a los/as estudiantes a comprender de forma intuitiva por qué combinar bloques de códigos con las funciones, puede ser una práctica muy útil.

## Propósito

El uso de funciones ayuda a simplificar el código y desarrollar la capacidad del estudiante para organizar su programa. Los estudiantes reconocerán rápidamente que las funciones de escritura pueden hacer que sus programas largos sean más fáciles de leer y más fáciles de realizar el debug si algo va mal.

## Agenda

### Actividad previa (20 min)

Vocabulario

Cantar una Canción

### Actividad Principal (20 min)

Funciones Sin Conexión: Componer Canciones - Ficha

### Actividad de Cierre (5 min)

Conversación Rápida: ¿Qué aprendimos?

Escribir en el Diario

### Evaluación (5 min)

Funciones Sin Conexión: Componer Canciones - Evaluación

### Aprendizaje Extendido

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Localizar frases repetidas dentro de las letras de las canciones.
- Identificar secciones de una canción para incorporar a una función.
- Describir cómo las funciones pueden hacer que los programas sean más fáciles de escribir.

## Preparación

- ▢Vea [teacher-video-songwriting-functions][0].
- ▢Vea [lesson-in-action-songwriting][1].
- ▢Imprima varias copias de [csf-songwriting-worksheets][2] para cada grupo.
- ▢Imprima una copia de [csf-songwriting-assessment][3] para cada estudiante.
- ▢Conéctese a internet o descargue previamente las canciones y letras para la actividad.
- ▢Asegúrese que cada estudiante tenga un [think-spot-journal][4].

## Enlaces

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

Para los docentes

- **Funciones Sin Conexión: Componer Canciones** - Video de lección en acción
- **Funciones Sin Conexión: Componer Canciones** - Guía de respuestas de la evaluación

Para los estudiantes

- **Componer Canciones con Funciones** - Video sin conexión (**descargar**)

- **Funciones Sin Conexión: Componer Canciones** - Ficha [Hacer una copia ▼](#)
- **Funciones Sin Conexión: Componer Canciones** - Evaluación [Hacer una copia ▼](#)
- **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.

# Guía docente

## Actividad previa (20 min)

### Vocabulario

Esta lección tiene una nueva e importante palabra:

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

Una parte de un código que puedes fácilmente usar repetidas veces.

### Cantar una Canción

- ¡Cuéntele a la clase que hoy es día de canciones!
- Vamos a aprender una canción juntos.
  - Comience con una canción simple, ya sea escrita o proyectada en la pantalla.
  - Apunte al estribillo y asegúrese que la clase sepa cómo es antes de cantarla.
  - Explore la canción, cantándola con ellos al principio, y luego vea lo que sucede cuando lleguen a la parte del estribillo.



#### **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*

*"I gave you two chances.  
Now I'll turn you into a goon!"  
(POOF!)*



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.

*And the moral of the story is:  
Hare today, goon tomorrow!*

- Es muy probable que la mayoría de la clase cante las letras para el coro cuando señale esa parte.
  - Detenga la canción una vez que eso suceda, y resalte explícitamente lo que acaba de suceder.
    - Usted definió el estribillo.
    - Usted llamó el estribillo.
    - Ellos cantaron el estribillo.
- Pregunte a la clase por qué creen 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 otros beneficios de escribir el estribillo una sola vez cuando se canta muchas veces?

Ahora, imagina que esta canción es un programa de computador. Definir un título (como "estribillo") para un pequeño trozo de código que 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 cometes un error, sólo tienes que cambiarlo a un lugar.
- El programa se siente menos complicado con las piezas repetitivas definidas una sola vez en la parte superior.

¡Vamos a jugar un poco más con las canciones, para tratar de entender realmente con qué frecuencia se utiliza esta técnica!

#### 💡 Recomendación para la lección

Para llegar a este punto de partida, puede buscar las letras de algunas canciones populares en Internet. Muestre a los/as estudiantes que el estándar para repetir la letra es definir el estribillo en la parte superior y llamarlo desde dentro del cuerpo de la canción.

## Actividad Principal (20 min)

### Funciones Sin Conexión: Componer Canciones - Ficha

Una forma fantástica de comparar las funciones con algo que vemos en nuestras vidas diarias, son las canciones. A menudo las canciones tienen ciertas partes de la letra que se repite varias veces. A esa parte le llamamos "estribillo".

#### Instrucciones:

- Divida a la clase en grupos de 4, 5, o 6.
- Entregue a cada grupo varias copias de la Hoja de Trabajo Componiendo Canciones.
- Ponga una canción corta que contenga un estribillo claro que no cambie de verso en verso.
- Desafíe al curso para que identifiquen (y anoten) el estribillo.
- Compare los resultados de cada grupo.
- ¿Todos obtuvieron lo mismo? ¡Canten sus estribillos a la misma vez para averiguarlo!

¿Lograron todos lo mismo? ¡Canten sus estribillos juntos para averiguarlo! 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 sobre lo que es el coro cantando juntos, en lugar de escribirlo todo. Si elige este método, considere que la clase haga un estribillo escrito para la selección final de la canción para asegurarse de que los/as estudiantes que aprenden mejor por medio de la vista, obtengan el refuerzo adecuado.

#### 💡 Sugerencias para la lección

Es más emocionante para los/as estudiantes hacer esta lección con música famosa de la radio, pero si le está costando encontrar canciones adecuadas donde las letras se repitan exactamente, aquí hay algunas alternativas:

- **You Are My Sunshine**
- **Boom, Boom, Ain't it Great**
- **How Much Is That Doggie in the Window**
- **I Love Trash**

## Actividad de Cierre (5 min)

## 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?
- ¿Qué es una función y cómo la usas?
- ¿Se te ocurre otra actividad donde pudieras necesitar usar un grupo de instrucciones varias veces?

## Evaluación (5 min)

### Funciones Sin Conexión: Componer Canciones - 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.

### Atrapasueños Funcionales

Visite el **CS Fundamentals Unplugged Table** o haga clic en el link **Functional Suncatchers**. Esta actividad ayuda a los/as estudiantes a ver el valor de llamar diferentes funciones.

### Crea Tu Canción

- Comiencen creando juntos un estribillo, luego repítanlo entre los versos de una canción que desarrollen alrededor de éste.
- Hagan un cambio en el estribillo, y evalúen cuanto más fácil es cambiarlo sólo en un lugar.
- Cambien el estribillo nuevamente, haciéndolo más largo de lo que era originalmente.
- Agreguen un segundo estribillo y altérnenlos entre los versos.

### Componiendo un Programa

- ¿Qué tal si actuamos las canciones en lugar de cantarlas? De repente, nuestros estribillos serán una función de acciones repetidas, más que palabras.
- Use el concepto de las flechas de la lección Programando con Papel Cuadrulado y creen un programa con muchas instrucciones repetidas.
  - Encierre en un círculo esas acciones repetidas, así la clase puede ver dónde están.
  - Defina una función llamada “Estribillo” al inicio del programa.
  - Haga una cruz en cada lugar que aparezcan las acciones repetidas en el programa y escriba “Estribillo” en su lugar.
- Repita hasta que la clase pueda pasar por este proceso con menos instrucciones.

# 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: Funciones en Artista

Función | Artista

## Descripción

Se introduce a los/as estudiantes al uso de las funciones en Code.org. Se crearán y modificarán imágenes extraordinarias con funciones en Artista. Para patrones más complejos, los/as estudiantes aprenderán acerca de anidar funciones llamando una función desde otra función.

## Propósito

Uno de los componentes más importantes de esta lección es proporcionar a los/as estudiantes un espacio para crear algo de lo que se sientan orgullosos/as. Estos desafíos progresan a imágenes más y más complejas, pero cada desafío sólo se construye a partir del desafío anterior. Al final de esta lección, los/as estudiantes sentirán confianza en ellos/as mismos/as y estarán orgullosos/as de su trabajo.

## Agenda

### Actividad previa (15 mín)

Introducción

### Actividad Puente - Funciones (15 mín)

Actividad Sin conexión Usando Algunos Blockly  
Pre visualización de los Desafíos Online

### Actividad Principal (30 mín)

Curso E Desafíos Online - 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

- ☐ Recorra los desafíos en **Course E Online Puzzles 2018 - Sitio web** para encontrar cualquier problema potencial 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 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

- **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.

# Guía docente

## Actividad previa (15 mín)

### Introducción

Pida a la clase que piensen en “Funciones Sin Conexión: Composición de Canciones” y que recuerden lo que es una función. Abra una discusión acerca de cuándo usar una función cuando escriben una canción.

Dígale a la clase que hay dos componentes principales al usar funciones.

1. **La Declaración:** Las Declaraciones de Función, son las que crean una función. En una declaración de función, completará la función con un código y le dará un nombre a la función. Debe declarar una función antes de usarla.
2. **El Llamado:** Las Llamadas de Funciones, son lo que hacen que el programa ejecute el código en la función. Para llamar a una función, ubica el nombre de la función en su programa. Asegúrese de que su función esté adecuadamente definida antes de llamarla a su programa.

La clase puede usar la composición de canciones como ejemplo para comprender estos dos componentes. En la actividad sin conexión, la función que contenía la letra del estribillo fue llamada “estribillo”. Cuando hicimos esta función por primera vez, encerramos en un círculo la letra que iría en la función. Una vez que nombramos la función, pudimos revisar la letra y reemplazar el estribillo repetido por una función llamada “estribillo”.

Continúe la conversación hasta que los/as estudiantes tengan una comprensión básica de la declaración y llamado de las funciones. Si ellos/as no llegan a este punto, asegúrese de hacer una de las Actividades Puente, antes de avanzar hacia los desafíos en Code.org.

## Actividad Puente - Funciones (15 mín)

Esta actividad ayudará a traer los conceptos sin conexión de “Componiendo canciones: Funciones Sin Conexión” al mundo online al que los/as estudiantes están ingresando. Escoja *una* de las siguientes actividades para realizar con su clase:

### Actividad Sin conexión Usando Algunos Blockly

Escoja una canción que los/as estudiantes disfruten e imprima copias de la letra. Puede usar la misma canción de “Funciones Sin Conexión: Composición de Canciones”. Separe a la clase en grupos o parejas. Distribuya la letra (incluyendo un estribillo repetido) y los bloques de función básicos de **Bloques Blockly Sin Conexión (Cursos 2-5) - Manipuladores** a cada grupo o pareja. Vea la sugerencia para la lección para mayor detalle.

Pídales que señalen cualquier parte de la canción que pueda ser transformada en una función (el estribillo es un buen ejemplo) y que lo ubiquen dentro de los bloques función proporcionados. Deberían completar la declaración de función con el nombre de la función y la parte repetida de la letra. Una vez que se ha hecho la declaración de función, pídales que llenen los llamados de función y que los ubiquen sobre las letras tachadas.

Una vez que cada grupo o pareja esté listo, pregúnteles dónde pusieron sus funciones y por qué. ¿Todos hicieron la misma función? ¿Qué tan a menudo se repite la función?

#### 💡 Sugerencia para la lección

Bloques Función:



El bloque de la izquierda es una declaración de función, un bloque que los/as estudiantes nombrarán y usarán para completar la función. El bloque de la derecha es un llamado de función, un bloque que hace que el código de la función se ejecute. Los/as estudiantes necesitarán varios de los bloques llamados de función.

## Pre visualización de los Desafíos Online

Seleccione un desafío desde el **Curso E Desafíos Online - Sitio web**. Recomendamos el primer desafío para esta actividad. Como clase recorran el desafío sin usar funciones. Una vez que tengan la solución, muéstrela en la pizarra o pantalla. Pregúntele a la clase que señalen al código repetido. Pregúnteles cómo simplificarían el programa.

En la pizarra o proyector, reescriba el programa sin el código repetido, pero dejando una línea de espacio. En esa/s línea(s), llame a la función. Al lado, declare la función como el bloque izquierdo de ejemplo en la sugerencia para la lección. Pregúnteles qué creen que el código hará ahora.

Comience una discusión con la clase sobre por qué las funciones podrían ser útiles en programación. Invítelos a discutir la diferencia entre las funciones y los loops.

## Actividad Principal (30 min)

### Curso E Desafíos Online - Sitio web

Los/as estudiantes podrían beneficiarse de escribir códigos sin funciones y luego crear funciones de los códigos repetidos. Si ellos/as no disfrutan haciendo esto en el espacio de trabajo de Code.org, recomendamos proporcionar papel y lápices para que ellos/as escriban (o dibujen) sus ideas.

## 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?
- ¿Cuáles son algunas de las diferencias entre las funciones y los loops?
- Haz un dibujo que hiciste hoy. ¿Puedes escribir el código necesario para crear esto?
- Haz un dibujo que te gustaría crear con código. Intenta escribir o redactar el código que haría ese dibujo.

## 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 17: Funciones en Abeja

Función | Abeja

## Descripción

En la segunda ronda de practica con las funciones online, los/as estudiantes navegarán por caminos complejos, recogerán suficiente néctar, y harán mucha miel.

## 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 (10 min)

Introducción

### Actividad Principal (30 min)

Course E Online Puzzles 2018 - Sitio web

### Actividad de Cierre (15 min)

Escribir en el Diario

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Usar funciones para simplificar programas complejos.
- Utilizar funciones predeterminadas para completar tareas que se repiten mucho.

## Preparación

- ☐ Recorra **Curso E Desafíos Online - Sitio web** para encontrar cualquier problema potencial 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 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

- **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.

# Guía docente

## Actividad previa (10 min)

### Introducción

Pregunte a los/as estudiantes si disfrutaron creando imágenes en Artista durante la última lección. Comience una discusión sobre los desafíos que encontraron (fáciles y difíciles).

Cuénteles que ahora usarán las funciones para simplificar el código en abeja. La abeja necesita atravesar un laberinto, recoger néctar, y hacer miel. Las funciones harán su programa más claro, compartimentando bloques complejos de código.

Este puede ser un buen momento para discutir las diferencias entre las funciones y los loops.

- Usas los loops cuando necesitas hacer algo varias veces, exactamente de la misma forma
- Usas una función cuando necesitas hacer algo en diferentes momentos en el mismo programa

## Actividad Principal (30 min)

### Course E Online Puzzles 2018 - Sitio web

En forma similar a la última lección, los/as estudiantes pueden beneficiarse de la programación sin funciones del código repetido. Recomendamos proporcionar papel y lápices para que escriban (o dibujen) sus ideas. También, si los/as estudiantes tienen problemas en reconocer patrones hágalos trabajar con un/a compañero/a en desafíos más difíciles.

## 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 hicieron tus funciones en los programas que escribiste hoy? ¿Cómo te ayudó eso?
- ¿Cuándo se debería utilizar una función en lugar de un loop?

## 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: Funciones en Cosechadora

Función | Cosechadora

## Descripción

Los/as estudiantes han practicado creando diseños maravillosos en Abeja, pero hoy usarán las funciones para cosechar los cultivos en Granjera. Esta lección los llevará a usar las funciones en formas nuevas combinándolas con los “loops mientras” y las afirmaciones “si/ si no”.

## Propósito

Esta lección está preparada para llevar a los/as estudiantes más allá en el uso de las funciones en formas más creativas. Usando también los condicionales y los loops, aprenderán que hay muchas formas de enfocar un problema, pero algunas son más eficientes que otras. Estos desafíos están hechos para mejorar las habilidades de resolución de problemas y el pensamiento crítico.

## Agenda

### Actividad previa (10 min)

Introducción

### Actividad Principal (30 min)

Course E Online Puzzles 2018 - Sitio web

### Actividad de Cierre (15 min)

Escribir en el Diario

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Reconocer cuándo una función podría ayudar a simplificar un programa.
- Utilizar funciones predeterminadas para completar tareas que se repiten mucho.

## Preparación

- ☐ Recorra **Curso E Desafíos Online - Sitio web** para encontrar cualquier problema potencial 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 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

- **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.



# Guía docente

## Actividad previa (10 min)

### Introducción

En este punto, sus estudiantes deberían ser introducidos a las funciones. Tómese este tiempo para que discutan las ventajas y desventajas de usar funciones en un programa. Hágalos discutir en parejas o como clase. Intenta usar ejemplos de desafíos difíciles y fáciles en Artista o Abeja.

Pregunte a la clase:

- ¿Cuándo usarías una función?
- ¿Por qué una función ayuda a simplificar tu programa?
- ¿Crees que las funciones hacen la programación más fácil o más difícil? ¿Por qué?

## Actividad Principal (30 min)

### Course E Online Puzzles 2018 - Sitio web

Algunos desafíos tendrán una función pre declarada para que los/as estudiantes completen. Puede ser útil para ellos/as escribir el programa completo sin una función primero, luego determinar dónde sería útil una función en el programa.

Es importante asegurarse que cada estudiante esté completando cada desafío con un punto verde oscuro. Si algunos/as de sus estudiantes tienen dificultades para simplificar el código y usar las funciones, establezca equipos de estudiantes expertos/as dentro de la clase para que caminen por la sala y respondan preguntas.

No se olvide de entregar lápices y papel a sus estudiantes para que puedan hacer bosquejos de posibles soluciones.

## 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 hace que te des cuenta de que una función podría ayudar a tu programa?
- ¿Cómo ayudan a tu programa los loops mientras y las afirmaciones si / si no ?

## 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: Determinar el Concepto

Abeja

## Descripción

Esta serie reúne conceptos de lecciones previas y le da a los/as estudiantes la oportunidad de pensar críticamente sobre como resolver cada problema, pero sin decirles que concepto deben aplicar. Los/as estudiantes revisarán algoritmos básicos, debugging, loops "repetir", condicionales, loops "mientras" y funciones.

## Propósito

Es importante para los/as estudiantes recordar que la Ciencia de la Computación proporciona suficientes oportunidades para ser creativos/as. Cada tema puede ser combinado con otro para hacer algo más grande y mejor. En esta lección, los/as estudiantes usarán juntos los conceptos aprendidos previamente, permitiendo una vista del "panorama general" de los proyectos de programación- Esta lección hará un puente entre las posibles brechas en la comprensión de cuándo usar ciertas herramientas de programación, en lugar de otras.

## Agenda

### Actividad previa (10 min)

Introducción

### Actividad Principal (30 min)

Course E Online Puzzles 2018 - Sitio web

### Actividad de Cierre (15 min)

Escribir en el Diario

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Reconocer el concepto de programación a utilizar para resolver un problema dado.
- Describir las diferentes formas en que se podría resolver un problema dado.

## Preparación

- ☐ Recorra el **Curso E Desafíos Online - Sitio web** asociado a este nivel para encontrar cualquier área potencial de problema para su clase.
- ☐ Revise **Sugerencias para la Actividad Principal de Fundamentos de Ciencias de la Computación - Recomendaciones de lecciones**.
- ☐ Asegúrese de 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

- **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

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

# Guía docente

## Actividad previa (10 min)

### Introducción

Esta lección está destinada a revisar conceptos previos mezclándolos.

Tómese este tiempo para pedir a los/as estudiantes que discutan acerca de sus partes favoritas del curso hasta ahora.

- ¿Qué les gustó aprender? ¿Por qué?
- ¿Qué no les gustó aprender? ¿Por qué?
  - Si fue muy difícil, ¿estarían interesados/as en hacer más actividades para comprender mejor esos conceptos?
- ¿Qué cosa genial tuvieron que hacer?
- ¿Qué esperan hacer con código algún día? ¿Qué tipos de herramientas de programación podrían usar para hacer eso?

Esta discusión debería ayudar a los/as estudiantes a entrar al marco del “panorama general” que esta lección está intentando enseñar. Pregúnteles cómo podrían usar en parejas, ciertas herramientas de programación como por ejemplo:

- Loops y condicionales
- Funciones y eventos

## Actividad Principal (30 min)

### Course E Online Puzzles 2018 - Sitio web

Debido a que los conceptos en este conjunto de lecciones no se entregan explícitamente, los/as estudiantes pueden enfrentarse a mayor frustración de lo habitual. Asegúrese de dar vueltas en la sala de clases y verificar que cada estudiante esté bien. Si un/a estudiante tiene alguna dificultad, emparejelo/a con otro/a estudiante para estimular la discusión entre ambos/as.

## 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?
- ¿Cuál es tu cosa favorita que has aprendido en tu clase de codificación hasta ahora? ¿Por qué? ¿Cómo has usado eso en los programas últimamente?

## 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: Construir un Juego de Play Lab

Evento | Laboratorio de juegos

## Descripción

Esta lección presenta Play Lab, una plataforma donde los/as estudiantes pueden crear sus propios juegos y tener interacciones entre los personajes y los aportes de los usuarios. Los/as estudiantes trabajarán con eventos para crear controles en el teclado. Este conjunto de desafíos guiará también a los/as estudiantes a través del desarrollo del juego, pero con libertad para agregar sus propias ideas.

## Propósito

Los/as estudiantes usarán eventos para hacer que los personajes de La Era del Hielo se muevan por la pantalla, muestren mensajes e interactúen con otros personajes según el usuario. Esta lección ofrece una entretenida introducción a los eventos en la programación, al mismo tiempo que proporciona la oportunidad de mostrar su creatividad. Al término de esta secuencia de desafíos, los/as estudiantes podrán compartir sus proyectos con sus amigos y familia.

## Agenda

### Actividad previa (10 min)

Introducción

Revisión de "Información Personal y Privada"

### Actividad Principal (30 min)

Course E Online Puzzles 2018 - Sitio web

### Actividad de Cierre (15 min)

Escribir en el Diario

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

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

## Preparación

- ☐ Navegue a través de **Course E Online Puzzles 2018 - Sitio web** asociado 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 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 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

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

# Guía docente

## Actividad previa (10 min)

### Introducción

Hoy, los/as estudiantes volverán a ver Eventos en Programación.

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

Pida a los estudiantes que levanten sus manos.

Lo que usted hizo fue declarar un evento. Cuando dice “levanten sus manos” los /as estudiantes responden levantando sus manos. En codificación, usted declararía esto diciendo “cuando diga “levanten sus manos” ustedes levantarán sus manos.

Puede también pensar en ciudades declarando eventos. Hay leyes que dicen “cuando hay una luz verde, los autos pueden pasar la intersección”. Pregunte a los/as estudiantes por qué creen que esto es un evento.

Hoy, los/as estudiantes jugarán en el Play Lab, pero los eventos con los que trabajarán serán más como videojuegos que están acostumbrados a jugar. Los eventos tomarán la forma de acciones, tales como presionar la flecha hacia arriba, o dos personajes corriendo hacia ellos.

### Revisión de “Información Personal y Privada”

Recuerde a los/as estudiantes la información que es segura para compartir online y la información que es estrictamente privada.

SEGURO - Información Personal	INSEGURO - Información Privada
Tu comida favorita Tu opinión (aunque debería ser dada con respeto) Nombre de Pila (con permiso)	Apellido de la madre Número de Carnet de Identificación Tu fecha de nacimiento Información de la tarjeta de crédito de los padres Número de teléfono

Discuta otros ejemplos de las dos categorías anteriores.

## Actividad Principal (30 min)



Recuérdale a los/as estudiantes compartir sus trabajos solamente con sus amigos/as cercanos/as o con la familia. Para más información vea o muestre a la clase [common-sense-media-pause-think][1].



### Course E Online Puzzles 2018 - Sitio web

Esta es una de las actividades online más libre del curso. En la etapa final los/as estudiantes tendrán la libertad de crear un juego propio. Puede entregarles guías estructuradas sobre qué tipo de juego hacer, en forma particular a los/as estudiantes que se sienten sobrepasados por tantas opciones.

## 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 qué se trata tu juego? ¿Cuáles son algunas de las características interesantes que tiene? ¿Qué tipo de código se necesitaba para hacer estas características geniales?
- ¿Qué es lo que crees que es realmente genial en los juegos que juegas? Cree una hipótesis sobre qué código sería necesario para crear esas características.

## 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: Explorar Ideas de Proyectos

Proyecto | Definir | Preparar | Probar | Reflexionar

## Descripción

Las próximas cuatro 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 contruidos 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 estudiantes podrán presentar sus trabajos terminados a sus compañeros 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)

Ejemplo de Proyectos

### Día 2 - El Proceso de Diseño (45 min)

Definir y Preparar

### Día 3. Construye Tu Proyecto (45 min)

Intentar

### Día 4 & 5 - Presenta Tu Proyecto (45 min cada uno)

Presentaciones

### Actividad de Extensión

Reflexionar e Intentar de nuevo (45 min.)

Reflexionar:

Otro

Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Aprender a planificar con anticipación para una asignación continua.
- 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

☐ Recorra el Proyecto en línea **Course E**

**Project - Examples** para tener ideas de las fortalezas, debilidades y limitaciones de la herramienta.

☐ Imprima una copia para cada estudiante de **Proceso de Diseño - Guía de preparación para el docente**.

☐ Modifique la **Proyecto Final de Fundamentos de Ciencias de la Computación - Rúbrica** para adaptarla a los objetivos de su clase e imprima 1 para cada estudiante.

☐ Modifique la **Diseño del Proyecto Final - Ficha** para adaptarla 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

- **Diseño del Proyecto Final - Ficha**

Hacer una copia ▾

- **Proyecto Final de Fundamentos de Ciencias de la Computación - Rúbrica**
- **Proceso de Diseño - Guía de preparación para el docente** Hacer una copia ▾

Para los estudiantes



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

Hacer una copia ▼

## 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)

### Ejemplo de Proyectos

Objetivo: Esta parte del proceso es de exploración. Los/as estudiantes se sentarán frente a un escenario lleno de ejemplos de proyectos para remezclar y aprender. Esto no sólo le dará una idea a los/as estudiantes de lo que es posible, también los /as ayudará a ver las limitaciones de la herramienta.

Dé a los/as estudiantes un día para jugar y remezclar los proyectos encontrados en **Course E Project - Examples**. Haga que usen sus diarios (o cuaderno) para llevar registro de los pensamientos e ideas mientras trabajan.

Esta actividad debe ser hecha en las mismas parejas o grupos con los que trabajarán los proyectos en las lecciones siguientes.

Asegúrese que su clase comprenda que pasarán las siguientes semanas trabajando en sus propios proyectos, así que deben poner mucha atención a cómo estos proyectos fueron escritos, así como a los conceptos que usan.

## 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.

## 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.

## Día 4 & 5 - 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

## Actividad de Extensión

### Reflexionar e Intentar de nuevo (45 min.)

Los/as estudiantes trabajarán con otro grupo para dar y recibir retroalimentación en un esfuerzo para fortalecerse mutuamente.

## Reflexionar:

Para las reflexiones, junte a cada grupo con otro para probar los proyectos del otro. Después de 10 minutos, haga que los grupos discutan las preguntas de la parte final de la Guía Diseño del Proyecto Final.

Anímelos/as a hacer las preguntas de la Guía Diseño del Proyecto Final y escriban la retroalimentación proporcionada por los grupos que revisaron el proyecto para así poder revisarlas más adelante. Esta parte debería tomar aproximadamente 15 minutos más.

Intentar de nuevo:

Con sus nuevas reflexiones escritas, los/as estudiantes pueden volver a los computadores para hacer las ediciones necesarias. Con sólo 10 minutos restantes ellos/as posiblemente tendrán que seleccionar sólo la retroalimentación más importante para incorporar.

### 💡 Recomendación para la lección

Los/as docentes deberían evitar asignar la parte final del proyecto como tarea a menos que estén seguros/as de que los/as estudiantes viven cerca unos de otros **y** que tienen internet en casa.

## Otro

Si sus estudiantes se sienten cómodos/as con los conceptos de codificación, intente que creen 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 22: El Proceso de Diseño

Proyecto

## Descripción

En el transcurso de cuatro lecciones los/as estudiantes se prepararán para programar un proyecto propio usando Play Lab o Artista como sus ambientes de programación. En esta parte del proyecto, los/as estudiantes aprenderán acerca del proceso de diseño y cómo implementarlo en sus propios proyectos. La guía de la lección para las 4 etapas del proceso puede ser encontrada en la primera etapa de este proceso del proyecto **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.

## Enlaces

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

Para los docentes

- **Proceso de Diseño** - Guía de preparación para el docente [Hacer una copia ▼](#)

# 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 23: Construye Tu Proyecto

Proyecto

## Descripción

En el transcurso de cuatro lecciones, los/as estudiantes se prepararán para programar un proyecto propio usando Play Lab o Artista como sus ambientes de programación. Ahora los/as estudiantes tendrán su propio espacio para crear su proyecto ya sea con Artista o Play Lab. Es posible que esta sea la etapa más larga del proyecto. La guía de la lección para las cuatro etapas del proceso, puede ser encontrada en la primera etapa de este 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 24: Presenta Tu Proyecto

Proyecto

## Descripción

En el transcurso de cuatro lecciones, los/as estudiantes construirán 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 cuatro 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 4 & 5 - 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 4 & 5 - 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.

# Lección 25: 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

Recibiendo 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** - Guía de respuestas de la evaluación [Hacer una copia](#)

### Para los estudiantes

- **Internet** - Video sin conexión ([descargar](#))
- **Tarjetas de dirección IP y tarjetas de tipo entrega - Manipuladores**  
[Hacer una copia](#)
- **Internet - Evaluación** [Hacer una copia](#)
- **Diario Think Spot - Diario de reflexiones**  
[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 distintas palabras importantes:

- **Dirección IP** - Díganlo conmigo: Di-rec-ción I-P

Un número asignado a cada ítem que está conectado a internet.

- **DNS (Domain Name Service)** - Díganlo conmigo: D-N-S

El servicio que traduce los URLs a direcciones IP.

- **URL (Universal Resource Locator)** - Díganlo conmigo: U-R-L

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

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

Un grupo de computadores y servidores que están interconectados.

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

Computadores que existen sólo para proveer cosas a otros.

- **Cable de Fibra Óptica** - Díganlo conmigo: Ca-ble de Fi-bra Óp-ti-ca

Una conexión que usa luz para transmitir información.

- **Wi-Fi** - Díganlo conmigo: Wai-Fai

Un método inalámbrico de enviar información usando ondas de radio.

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

Una forma de enviar información usando el teléfono o cables de televisión.

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

Pequeños bloques de información que han sido hechos cuidadosamente de bloques de información más grandes.

### Recibiendo el Mensaje

- Es muy probable que sus estudiantes sepan lo que es internet, pero es posible que no comprendan lo que hace internet.

- Pregunte "¿Qué es internet?"
- ¿Es internet un espacio público o privado?
- (A decir verdad, mucha gente piensa que pueden ser ambos, debería ser visto como un lugar público sin importar lo que creas que dominas).
- ¿Cómo viaja la información de lugar en lugar?
- Digamos que quiero mirar una página en [Code.org](http://Code.org)
  - ¿Cómo cree que sería el proceso para que yo enviara un mensaje para requerir esa página?
  - ¿Qué hago como usuario?
  - ¿Qué crees que pasa al interior de internet?

Enviar un mensaje por internet es muy similar a enviar un mensaje por correo ¡si cada carta que enviamos requiere miles de sobres!

#### 💡 Recomendación para la lección

Aquí solo es necesario una revisión rápida. Estas palabras serán explicadas como parte de la lección, así que será menos confuso hacer una introducción breve a las palabras solo para ver si "pueden reconocerlas en la vida diaria".

#### 💡 Sugerencia para la lección

Hay unos videos grandiosos en YouTube sobre este tema que pueden hacer esta lección un poco más fácil de comprender. Puede mostrarlos a la clase por adelantado, o solo verlos usted. **Aquí hay uno de los videos en [Code.org](http://Code.org) sobre "Cómo funciona Internet".** (Recomendamos ver desde 1:44 - 5:13, si es posible). El resto de la lista de reproducción está disponible en [here](#).

Cada mensaje que enviamos por internet es separado en partes pequeñas, cada pieza se envuelve en su propia versión de un sobre. A eso le llamamos “paquetes”. Los paquetes están especialmente formados por bloques de información que puede viajar fácilmente a través de cualquier canal de internet.

A veces, algunos de esos paquetes se perderán, porque internet es un lugar loco. En ese caso, los paquetes necesitan ser re enviados y todo el mensaje tiene que ser puesto en espera hasta que ellos lleguen.

¿Hacia dónde crees que se dirigen esos paquetes?

- Incluso si estás enviando mensajes a otra persona, primero tienen que ir al menos a un “servidor”.
  - Un servidor es un computador especial que se supone debe estar siempre encendido y listo para enviar y recibir información.
  - Cada sitio web tiene un servidor.
  - Incluso los emails van a través de servidores.

Los servidores no tienen nombre como tú o como yo. En realidad sus nombres son unos números. Estos números se llaman direcciones IP, y se ven un poco extraños.

- Por ejemplo: una dirección de IP que Code.org solía usar es 54.243.71.82
  - (Por favor asegúrese de revisar esto antes. La mayoría de las direcciones IP cambian de vez en cuando y se re usan para otros sitios).

Hay muchas formas de conectarse a internet desde tu casa, escuela o trabajo.

- Puedes conectarte directamente usando un cable (que puede ser: DSL, Cable, o Fibra Óptica).
- O puedes conectarte usando ondas de radio por el aire por medio del Wi-Fi.

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

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

Las conexiones Wi-Fi son súper convenientes, pero no son siempre confiables.

- ¿Puedes descubrir por qué no?
  - (Las ondas de radio rebotan por todos lados y se pueden perder.)

[www.code.org]

Así que, si estás acostumbrado/a a enviar información a URLs (como **www.code.org**) y los servidores tienen realmente direcciones IP como nombre (como 54.243.71.82) ¿Cómo internet cambia de uno a otro? Para eso es el DNS. El DNS (Domain Name Server) tiene tablas que permiten al sistema ir y regresar entre URLs y direcciones IP. Si el Domain Name Server deja de trabajar alguna vez, ¡se acabaría internet como lo conocemos!

Dicho eso, intentemos comprender lo que hace un DNS haciendo una pequeña tabla DNS nosotros mismos.

Ejemplo de tabla DNS:

#	URL	Dirección IP
1	code.org	54.243.71.82
2		
3		

#### 💡 Recomendación para la lección

Si cree que esto es mucho texto y que sería muy aburrido intentar leer esto a la clase llena de estudiantes de básica, ¡está absolutamente en lo correcto! Si no puede mostrar un video de You Tube en clases para ayudar a la explicación, recomiendo hacer dibujos para explicar cada idea, o escoja estudiantes como voluntarios para actuar lo que lo que usted describe mientras lo explica. No se espera que entiendan cada detalle y definiciones en este punto, solo recibir exposición al tema.

#	URL	Dirección IP
4		
5		

Primero, necesitamos completar esta tabla.

- Haga una pequeña encuesta con la clase para saber sus sitios webs favoritos y escriba las URLs en la columna izquierda.
- Use un sitio como **get-site-ip.com** para encontrar las direcciones IP de esos sitios y escríbalas en las filas correspondientes en la columna de la derecha.

¡Ahora tomemos esta tabla DNS y simulemos que enviamos un mensaje a través de 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 todos 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).

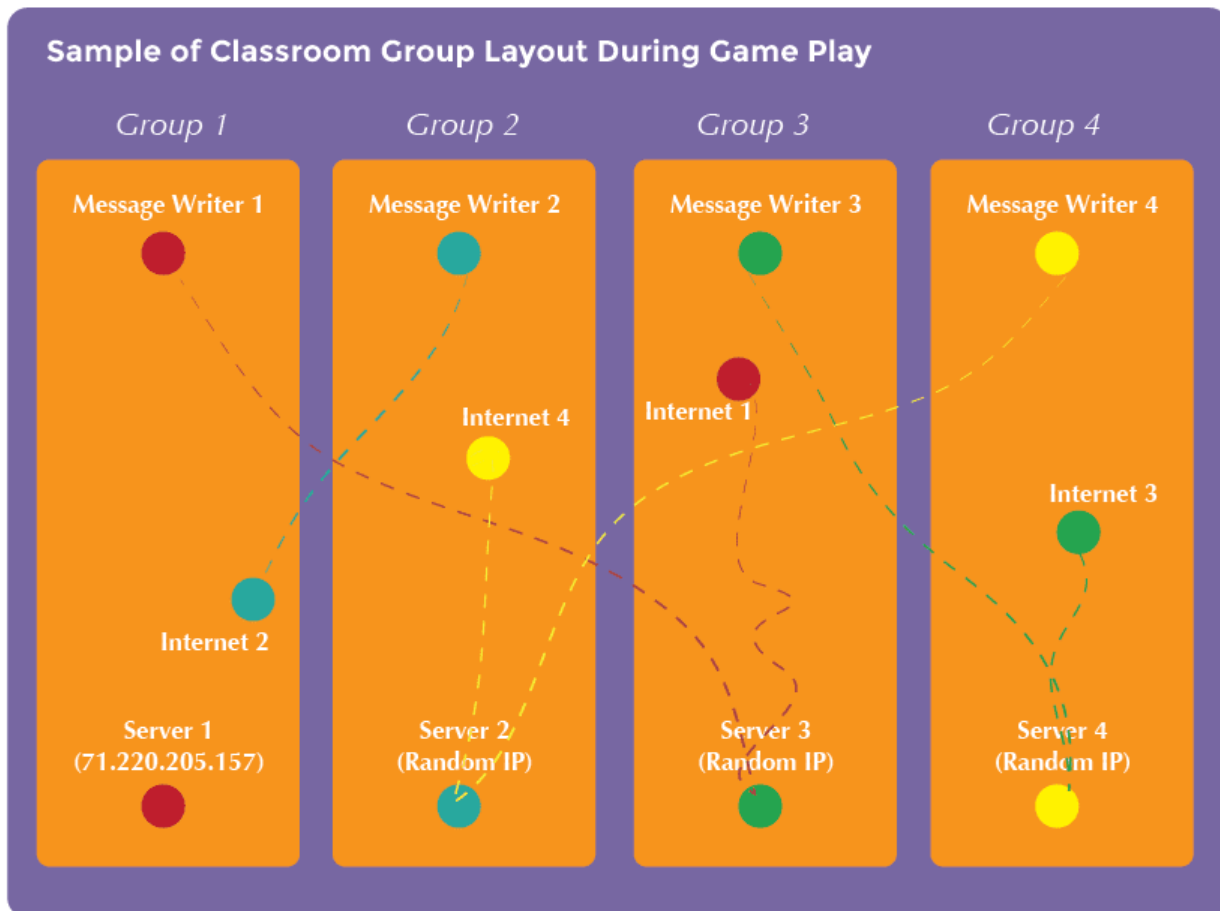
- Cable/DSL: Bastante Buena para entregar mensajes, pero debe estar conectado a un cable.
  - Simulación Internet debe llevar cada paquete en el dorso de la mano y con la otra mano debe estar tocando siempre un muro, escritorio, silla el suelo.
- Cable de Fibra Óptica: El mejor para entregar mensajes, 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 26: Más Allá de la Programación: Crowdsourcing

Sin conexión | Más Allá de la Programación: Crowdsourcing

## Descripción

En la ciencia de la computación, enfrentamos algunos problemas abrumadores. Los desafíos como encontrar grandes números primos o secuencias de ADN son casi imposibles de hacer individualmente. Al unirse con el poder de otros, estas tareas se hacen más manejables. Esta lección mostrará a sus estudiantes lo útil que es el trabajo en equipo en la industria de la ciencia de la computación.

## Propósito

Es muy extraño que un científico de la computación trabaje completamente solo en un proyecto. Incluso cuando eso pasa, hay siempre un beneficio en los números. Hoy, los/as estudiantes aprenderán lo que significa hacer crowdsourcing con un proyecto. Esta actividad construye el trabajo en equipo y crea un entorno eficiente para que los/as estudiantes resuelvan problemas.

## Agenda

### Actividad previa (20 min)

Vocabulario

Introducción

### Actividad Principal (20 min)

Crowdsourcing - Ficha

### Actividad de Cierre (15 min)

### Conversación Rápida: ¿Qué aprendimos?

Escribir en el diario

### Aprendizaje Extendido

## Ver en Code Studio

## Objetivos

Los estudiantes podrán:

- Identificar una tarea grande que debe llevarse a cabo.
- Reorganizar una tarea grande en varias tareas más pequeñas.
- Crear una solución completa a partir de varias soluciones más pequeñas.

## Preparación

- ☐ Vea el **Más Allá de la Programación: Crowdsourcing - Video para el docente**.
- ☐ Revise **Crowdsourcing - Ficha**.
- ☐ Tenga un tarro con grandes cantidades de algo (monedas, botones, trozos de papel, etc.) y un mazo de cartas.
- ☐ Asegúrese de 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

- **Más Allá de la Programación: Crowdsourcing** - Video sin conexión (descargar)
- **Crowdsourcing** - Ficha

Para los estudiantes

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

Hacer una copia ▾

## Vocabulario

- **Más Allá de la Programación: Crowdsourcing** - obtener ayuda de un gran grupo de personas para terminar algo más rápido.

# Guía docente

## Actividad previa (20 min)

### Vocabulario

Esta lección tiene una nueva e importante palabra:

**Crowdsourcing** - Díganlo conmigo: Crowd-sour-cing

Obtener ayuda de un gran número de personas para terminar algo más rápido.

### Introducción

- Muestre a sus estudiantes su frasco lleno de algo.
  - "Miren este frasco. Tengo muchos botones aquí y necesito decirle al/a director/a cuántos hay antes que termine la clase".
  - "¿Se les ocurre una forma en que podría contarlos rápidamente?"
- Sus estudiantes puede que lo/a lleven a pedir ayuda, pero si no lo hacen, lo puede sugerir usted.
  - Vierta todos los botones (o las monedas, etc.) en un montoncito en el piso.
  - Invite a los/as estudiantes a acercarse y tomar un pequeño número (10 está bien, pero pueden ser más si sus estudiantes lo pueden manejar).
  - Una vez que hayan contado sus 10, pídales que le reporten, vuelvan a poner los botones en el frasco y háganlo de nuevo hasta que el montoncito esté vacío. • Comente lo rápido que se realizó la tarea.
  - Haga que la clase reflexione sobre lo difícil que se puede haber sentido hacerlo solo/a.

#### 💡 Sugerencia para la lección

Los frascos de botones y monedas funcionan bien, pero si tiene un poco de tiempo para preparar, puede cortar tiras de papel y ponerlas en una bolsa ziplock o incluso un estuche.

## Actividad Principal (20 min)

### Crowdsourcing - Ficha

A veces tienes un trabajo enorme que hacer, pero se siente como si fuera a durar eternamente. Crowdsourcing es una forma de usar el trabajo en equipo, para ayudar a que este se haga más rápido. En este juego, usaremos crowdsourcing para clasificar mazos de cartas.

#### Instrucciones

Sugerencia para la lección[1]

Puede ser un gran desafío para los/as estudiantes descifrar como separar tareas grandes al principio. Los/as estudiantes podrían encontrar útil recibir algunas ideas después de trabajar un rato. Una división buena para repartir las cartas es esta:

- Una persona recoge las cartas y determina la pinta para cada uno.
- Una persona maneja los Corazones.
- Una persona maneja los Diamantes.
- Una persona maneja los Tréboles.
- Una persona maneja las Espadas.
- (Si hay otro, pueden volver a juntar todas las pintas)

[/][1]

1. Divídalos en grupo de 4, 5, ó 6.
2. Tome su mazo de cartas y póngalo dentro de una bolsa, balde o incluso en un bolsillo hecho con la parte de abajo de su camisa.
3. Agite las cartas hasta que estén todas revueltas.

4. Ponga las cartas sobre una mesa o escritorio donde todo el grupo las pueda ver.
5. Decida como dividir la tarea de repartir las cartas, para que cada persona tenga algo que hacer y nadie tenga que hacer demasiado.
6. Toma el tiempo mientras repartes las cartas. ¿Se le ocurre una manera de hacerlo más rápido?
7. Repita el juego varias veces hasta que hayan encontrado la manera más rápida de hacer crowdsourcing con la actividad de repartir cartas.

## Actividad de Cierre (15 mín)

### Conversación Rápida: ¿Qué aprendimos?

- ¿Has intentado separar un montón de cartas por pinta solo/a?
- ¿Crees que fue más fácil o más difícil con ayuda?
- ¿Qué otras cosas tienes que hacer en la vida que pueden ser más fáciles con ayuda?

#### 💡 Sugerencia para la lección

Las preguntas de la Conversación Rápida buscan ver la imagen completa para ayudar a pensar cómo la lección se relaciona con el mundo y el futuro de los/as estudiantes. Use su conocimiento y 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

- ¿De qué se trató la lección de hoy?
- ¿Cómo te sentiste durante la lección de hoy?
- ¿Cuáles son los beneficios del crowdsourcing?
- ¿Qué tipo de cosas quieres hacer con ciencia de la computación? ¿Cómo ves que el crowdsourcing pueda ser beneficioso en esos proyectos?

## Aprendizaje Extendido

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

### Crowdsourcing inverso

A menudo vemos al crowdsourcing como separar cosas para hacerlas más simples. También puedes hacer cosas grandes, hermosas con la misma técnica.

Pida a sus estudiantes que cada uno/a escoja 3 cartas y construya una sección de una **casa de naipes**. Cada estudiante puede ir uno después del otro para construir una gran torre de naipes.

Intente con 2 o incluso 3 estudiantes, agregando sus bloques uno a la vez.

- ¿El crowdsourcing siempre hace una tarea más fácil?

### Crowdsourcing en el camino

- Puedes hacer crowdsourcing con todos o puedes hacerlo con una persona a la vez. Intente tener a toda la clase clasificando el mismo mazo de cartas, un/a estudiante la vez.
  - Revuelva las cartas y ubíquelas en un montón en el centro de la sala.
  - Pida a los/a estudiantes que se acerquen y escojan 4 cartas.
    - Tenga 4 montones para que los/as estudiantes clasifiquen sus cartas.
      - Espada
      - Tréboles
      - Corazones
      - Diamantes
    - Una vez que todas las cartas han sido ubicadas en los 4 montones, haga que los 4 estudiantes siguientes clasifiquen los montones individuales.
    - El último pondrá los 4 montones juntos.
- Esta versión no ahorra mucho tiempo, pero divide el trabajo y le deja a cada individuo más tiempo libre!

# 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.