

## Lección 2

# Introducción a los desafíos en línea

### Propósito

En este conjunto de desafíos, los estudiantes comenzarán con una introducción (o repaso, dependiendo de la experiencia de su clase) del espacio de trabajo en línea de Code.org. Habrá videos indicando las funciones básicas del espacio de trabajo, como los botones ejecutar (run), reiniciar (reset) y paso (step). En estos videos también se aborda cómo arrastrar, borrar y conectar bloques Blockly. Luego, sus estudiantes pondrán en práctica sus habilidades de secuenciación y depuración en un laberinto. A partir de ahí, los estudiantes verán nuevos tipos de desafíos, como la recolectora, el artista o la cosechadora, mientras aprenden lo básico de los bucles.

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

### Secuencia para el aprendizaje

Conocimiento inicial (10 min)

Lección puente: programación (10 min)

Ampliación del conocimiento (30 min)

Transferencia del conocimiento (10 min)

### Objetivos

Los estudiantes serán capaces de:

Lección en línea  
[Ver en Code Studio](#)

### Recursos

#### ¡Atención!

Por favor, haga una copia de cada documento que planea compartir con los estudiantes.

Para los estudiantes:

- Video – [Pair programming.](#)
- Puntos clave – [Programación en parejas.](#)

### Vocabulario

- **Bug (error):** parte de un programa que no funciona correctamente.
- **Depurar (debugging):** encontrar y solucionar los problemas en un algoritmo o programa.
- **Bucle (loop):** la acción de hacer algo una y otra vez.
- **Programa:** algoritmo que ha sido codificado de forma que pueda ser ejecutado por una máquina.
- **Programación:** el arte de crear un programa.

- Ordenar comandos de movimientos como pasos secuenciales de un programa.
- Modificar un programa existente para reparar errores.
- Descomponer una secuencia larga de instrucciones en una secuencia de repetición más corta.

### Preparación

- Realice los desafíos para encontrar cualquier área potencialmente problemática para su clase.
- Asegúrese de que cada estudiante tenga su [Bitácora](#).

## Estrategia de aprendizaje

### Conocimiento inicial (10 min)

#### Introducción

Los estudiantes aprenderán muchos conceptos nuevos o repasarán muchos conceptos básicos. En base a la experiencia de su clase, puede revisar las palabras programa, programación, bug, depurar, ciclo o bucle en el glosario del curso o pasar directamente a la Lección puente. Si las definiciones no están cubiertas explícitamente, recomendamos usar las palabras en oraciones.

### Lección puente: programación (10 min.)

Esta Lección ayudará a sus estudiantes a traer los conceptos sin conexión de “programación con papel cuadriculado” al mundo digital al cual se están sumergiendo.

#### Revisión de desafíos en línea en conjunto

Elija un desafío de la Lección. Para esta Lección recomendamos progresar con [estos desafíos](#). Divida a los estudiantes en grupos de tres o cuatro. Pídales que programen a “Red”, de Angry Birds, para que llegue al cerdo, usando las flechas de “programación con papel cuadriculado”.

La clase no necesitará usar el último símbolo.



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

### Ampliación del conocimiento (30 min.)

#### Desafíos en línea

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

- Use la programación en parejas cada vez que sea posible durante la Lección.
- A través de desafíos o preguntas, anime a los estudiantes para que busquen respuestas con sus respectivas parejas.
- Las preguntas sin responder pueden ser delegadas a un grupo cercano, que podría ya tener la

respuesta.

- Recuérdelos usar el proceso de depuración antes de que usted se acerque a ayudar.
- Pida a los estudiantes que describan el problema que estén viendo. ¿Qué se supone que debe hacer?, ¿qué hace?, ¿qué te dice eso?
- Recuérdelos que la frustración es un paso en el camino del aprendizaje y que la perseverancia dará sus frutos.
- Si un estudiante sigue con problemas para avanzar después de todo esto, haga preguntas clave para que los estudiantes identifiquen el bug (error) por ellos mismos.

### Lección en Code Studio

Acceden a la plataforma para realizar los desafíos en línea.

### **Transferencia del conocimiento (10 min)**

#### Escribir en la bitácora

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

Sugerencias para la bitácora:

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

### **Sugerencias de Evaluación**

Se sugiere evaluar formativamente los aprendizajes:

- Resuelven problemas
- Asignan roles en los grupos
- Dividen un problema en otros menores

Enséñeles a los estudiantes la manera correcta de trabajar en parejas:

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