

Programación para todos Preescolar



Guía para profesores



Índice

Introducción

Comandos

- Lección 1: Rutinas diarias
- Lección 2: Orden de las historias
- Lección 3: Pasos de baile

Funciones

- Lección 1: Gema de papel
- Lección 2: Festival de música
- Lección 3: Mi función relajante

Bucles

- Lección 1: Pétalos que se repiten
- Lección 2: Carrera de obstáculos
- Lección 3: Patrones de percusión

Variables

- Lección 1: Hundirse o flotar
- Lección 2: Juego de palabras
- Lección 3: Todo sobre mí

Diseño de apps

Recursos para los instructores

«*Programación para todos — Preescolar*» está diseñado para ayudar a los educadores y las familias a introducir la programación en los primeros cursos, cuando los alumnos están comenzando a desarrollar habilidades de pensamiento computacional. A través de estas lecciones, los alumnos de 5 a 8 años explorarán, descubrirán y jugarán para sentar las bases de los conceptos básicos de programación.



Diseño de material de formación

Esta guía se divide en cuatro módulos y en un proyecto final de diseño de apps. Cada módulo contiene tres lecciones, cada una de las cuales se centra en un concepto relacionado con la programación. Dentro de cada lección, encontrarás tres actividades: Explorar, Descubrir y Jugar. Las actividades se pueden dividir en varias sesiones o días.

Día 1: debate y aprendizaje práctico

Explorar

Presentar y comentar el concepto de programación

Descubrir

Familiarizarse con el concepto mediante actividades creativas

~25
minutos

Jugar

- Programa junto con Byte en la app Swift Playgrounds
- Practica la programación en hojas de trabajo complementarias y actividades de Keynote
- Lleva el mundo de Byte al mundo real con los juegos de programación de rompecabezas en el suelo, sin pantallas

~25
minutos

Día 2: relacionar el aprendizaje con el código

Ámbito y secuencia

Los cuatro módulos de esta guía están diseñados para su uso con alumnos de 5 a 8 años y se pueden realizar en cualquier orden. Te recomendamos que utilices el módulo Diseño de apps en cualquier momento o incluso varias veces al año a medida que los alumnos comprendan mejor el código y las apps.

Ejemplo:

Edad	Módulo	Proyecto final	Tiempo total aproximado
5 años	Comandos	Diseño de apps	4 horas
6 años	Funciones	Diseño de apps	4 horas
7 años	Bucles	Diseño de apps	4 horas
8 años	Variables	Diseño de apps	4 horas

Seguir aprendiendo

Para la enseñanza a alumnos de 9 a 14 años, el plan de estudios «Programación para todos – Rompecabezas», junto con el diario de diseño de apps y la guía de presentación de apps, ofrece más de 45 horas de aprendizaje. Más información sobre la [guía del plan de estudios «Programación para todos»](#).



Carpetas de trabajo de los alumnos (opcional)

A lo largo de estos módulos, recopila artefactos de las actividades para crear carpetas de trabajo con tus alumnos.



Módulo	Lección	Artefactos sugeridos
Comandos	Rutinas diarias	<ul style="list-style-type: none"> • Hoja de trabajo «Dar órdenes con comandos» • Hoja de trabajo «Añadir un comando nuevo»
	Orden de las historias	<ul style="list-style-type: none"> • Imagen de puntos de la trama «Orden de las historias» • Imagen de grupo de «Orden de las historias»
	Pasos de baile	<ul style="list-style-type: none"> • Tarjetas de «Pasos de baile» • Vídeo de «Pasos de baile» (opcional)
Funciones	Gema de papel	<ul style="list-style-type: none"> • Forma de gema de papel • Hoja de trabajo «Componer un comportamiento» • Hoja de trabajo «Crear una función nueva»
	Festival de música	<ul style="list-style-type: none"> • Función escrita o vídeo de un concierto de «Festival de música»
	Mi función relajante	<ul style="list-style-type: none"> • Dibujo o vídeo de «Mi función relajante» • Hoja de trabajo «Recoger, activar y repetir»
Bucles	Pétalos que se repiten	<ul style="list-style-type: none"> • Pétalos que se repiten • Hoja de trabajo «Utilizar bucles» • Hoja de trabajo «Bucles por los cuatro costados»
	Carrera de obstáculos	<ul style="list-style-type: none"> • Vídeo o imágenes de la carrera de obstáculos (opcional)
	Patrones de percusión	<ul style="list-style-type: none"> • Hoja de trabajo «Idea y vuelta» • Vídeo o imágenes de «Patrones de percusión» (opcional)
Variables	Hundirse o flotar	<ul style="list-style-type: none"> • Hundirse o flotar • Hoja de trabajo «Llevar el recuento»
	Juego de palabras	<ul style="list-style-type: none"> • Juegos de palabras
	Todo sobre mí	<ul style="list-style-type: none"> • Todo sobre mí • Todo sobre ti
Diseño de apps		<ul style="list-style-type: none"> • ¿Qué es una app? • Mi diseño de app • Prototipo de diseño de apps

Primeros pasos con Swift Playgrounds en el iPad o el Mac



Antes de adentrarte en las lecciones, asegúrate de haber descargado [Swift Playgrounds](#), [Pages](#) y [Keynote](#).

Los módulos de esta guía utilizan diferentes combinaciones de áreas de juegos. Esto es lo que necesitarás para cada módulo:

Módulo	Áreas de juegos	Cómo descargarlas en Swift Playgrounds
Comandos	 Aprende a programar 1  Danzas MeeBot	<p>Para suscribirte a los contenidos de MeetBot de Playgrounds, desplázate hasta la parte inferior de la pantalla «Más áreas de juegos», toca la URL «Introducir una suscripción» e introduce lo siguiente: ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json.</p>
Funciones	 Aprende a programar 1	
Bucles	 Aprende a programar 1  Danzas MeeBot	<p>Para suscribirte a los contenidos de MeetBot de Playgrounds, desplázate hasta la parte inferior de la pantalla «Más áreas de juegos», toca la URL «Introducir una suscripción» e introduce lo siguiente: ubtechrobotics.github.io/MeebotPlaygroundFeed/locales.json.</p>
Variables	 Aprende a programar 2  Piedra, papel o tijera  Máquina de código	<p>Puedes encontrar «Piedra, papel o tijera» y «Máquina de código» en la sección «Libros» de la pantalla «Más áreas de juegos».</p>
Diseño de apps		

Consulta los requisitos mínimos de Swift Playgrounds en el [App Store](#). Visita el [soporte técnico de Apple](#) para obtener ayuda con Swift Playgrounds.

Consejos para los instructores

Para aprovechar al máximo las lecciones con tus alumnos, prueba algunos de estos consejos.

Actividades Explorar y Descubrir:

- Simplifica cualquier sintaxis o mayúsculas y minúsculas al escribir o mostrar código, por ejemplo:
 - `var names = ["Rose", "Sam", "Joy"]` --> `var names = Rose, Sam, Joy`
 - `var ages = [7, 8, 7, 8, 7]` --> `var ages = 7, 8, 7, 8, 7`
 - `var myFavoriteColor = ■` --> `var my favorite color = ■`

Actividades Jugar:

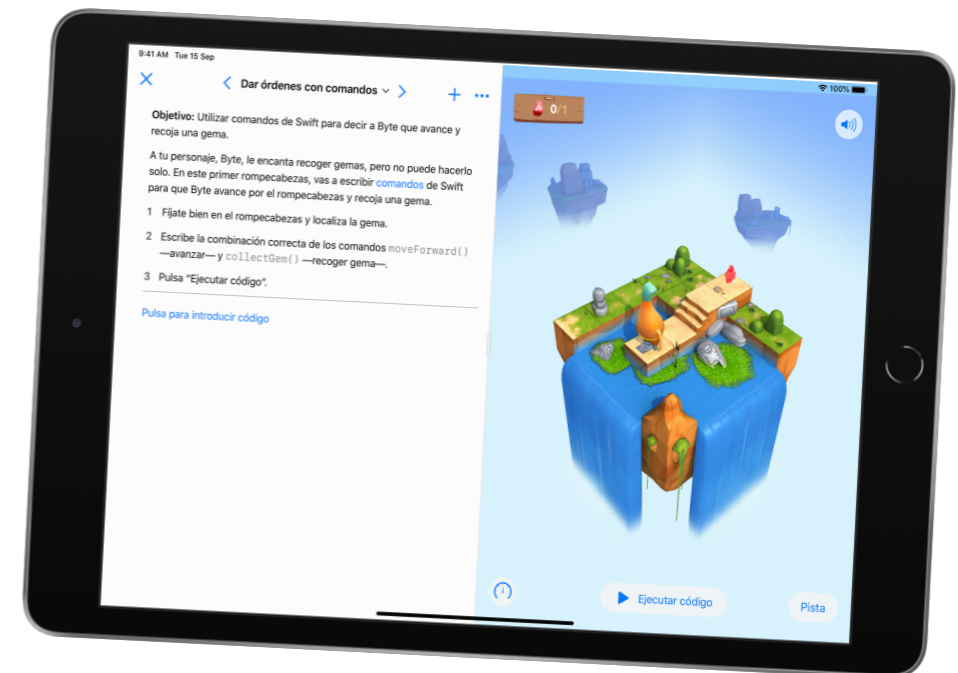
- Para que la app Swift Playgrounds sea aún más sencilla para los alumnos de preescolar, sigue las instrucciones incluidas en los planes de lecciones. Estas son:
 - Lee las presentaciones en grupo
 - Proporciona a los alumnos instrucciones detalladas sobre las hojas de trabajo adjuntas para que puedan encontrar sus propias soluciones
 - Usa un iPad o Mac de instructor para resolver los rompecabezas de la app
- `let` y `var`: la palabra clave `let` no se trata en esta guía. Para evitar confusiones en Swift Playgrounds, cambia todas las palabras clave `let` a `var` antes de mostrar las páginas a los alumnos. En las áreas de juegos que recomendamos, las dos palabras clave son intercambiables.
 - `let` = la variable no cambia
 - `var` = la variable cambia

Ampliaciones de la actividad:

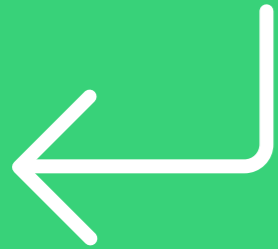
- Amplía las actividades de «Juego en el suelo» para que incluyan aritmética, lectura y escritura, palabras ilustradas, ortografía y mucho más. Prueba la actividad de «Juego en el suelo» del módulo «Funciones» para inspirarte.
- Personaliza las actividades de «Juego en el suelo» haciendo que los alumnos creen sus propias tarjetas para los comandos, como `twirl()` o `jump()`.



Página de introducción



Página de área de juegos



Comandos



Descripción

Lección 1: Rutinas diarias

- Explorar: debate que relaciona la repostería con los comandos
- Descubrir: actividad «Rutinas diarias»
- Jugar: «Dar órdenes con comandos» y «Añadir un comando nuevo»

Lección 2: Orden de las historias

- Explorar: debate que relaciona el orden de las tramas de las historias con los comandos
- Descubrir: actividad «Orden de las historias»
- Jugar: juego de rompecabezas en el suelo

Lección 3: Pasos de baile

- Explorar: debate que relaciona los pasos de baile con los comandos
- Descubrir: actividad «Pasos de baile»
- Jugar: «Hola, MeeBot» y «Movimientos básicos»

Los alumnos podrán

- Usar ejemplos cotidianos para describir instrucciones paso a paso
- Poner las instrucciones en orden para que tengan sentido
- Probar y depurar las instrucciones y el código

Vocabulario

- **Secuencia:** el orden en el que suceden las cosas
- **Paso:** una acción en un proceso más amplio
- **Modificar:** cambiar
- **Comando:** código que indica a una aplicación que realice una acción específica
- **Error:** un error de código
- **Depurar:** encontrar y corregir errores de código

Explorar

Objetivo: introduce el concepto de comandos relacionándolo con hornear brownies.

Debate:

- ¿Seguirían una receta de brownie?
- ¿Seguirían los pasos de la receta en orden?

Idea clave: cada paso o instrucción de una receta es como un comando en código. Pide a los alumnos que creen sus propios comandos.

Descubrir

Objetivo: representa el proceso de una rutina diaria identificando instrucciones paso a paso.

Materiales: tarjetas «Lavarse las manos»

Indicaciones:

1. Baraja el mazo de tarjetas «Lavarse las manos» y colócalas sobre una mesa o en la pizarra. Las tarjetas deben estar desordenadas.
2. Pregunta a los alumnos si creen que hay un error en la secuencia de lavado de manos.
3. Pide a los alumnos que depuren (o corrijan) las instrucciones moviendo las tarjetas de una en una a su ubicación correcta.

Alternativa:

Pide a los alumnos que trabajen en parejas o en grupos pequeños y entrega a cada grupo un juego de tarjetas.

Ampliación de la actividad:

Pide a los alumnos que creen su propio conjunto de instrucciones paso a paso sobre algo que hagan todos los días y que hagan dibujos de los pasos específicos.

↓ [Descarga las tarjetas «Lavarse las manos»](#)



Jugar

Objetivo: los alumnos podrán añadir los comandos en el orden correcto para recoger sus primeras gemas en «Aprende a programar 1» en la app Swift Playgrounds.

Indicaciones:

1. Proyecta la página de introducción del capítulo «Comandos» del área de juegos «Aprende a programar 1» en una pantalla.
2. Introducción:
 - Lee las páginas con toda la clase y haz preguntas si es necesario.
3. Dar órdenes con comandos:
 - Revisa los dos comandos que los alumnos necesitarán para que Byte llegue a la gema, `moveForward()` y `collectGem()`.
 - Pide a los alumnos que experimenten formas de dirigir a Byte desde la flecha de inicio hasta la gema para recogerla. Pueden registrar los comandos en la hoja de trabajo o en una hoja de papel aparte.
 - Recopila ideas de la clase y escribe el código en la app Swift Playgrounds para completar el rompecabezas. Haz clic o toca Ejecutar código.
 - Prueba varias ideas diferentes.
 - ¡Celébralo con Byte!

Ampliación de la actividad:

Si los alumnos están listos, pasa a la página siguiente, «Añadir un comando nuevo». Aquí, los alumnos usarán un nuevo comando, `turnLeft()`.




Aprende a programar 1

Materiales para los instructores:

- iPad o Mac
- App Swift Playgrounds
- Área de juegos «Aprende a programar 1»
- Proyector o pantalla

Materiales para los alumnos:

- Hojas de trabajo «Dar órdenes con comandos» y «Añadir un comando nuevo»
- Lápices
- Papel adicional (opcional)


[Descarga las hojas de trabajo «Aprende a programar»](#)



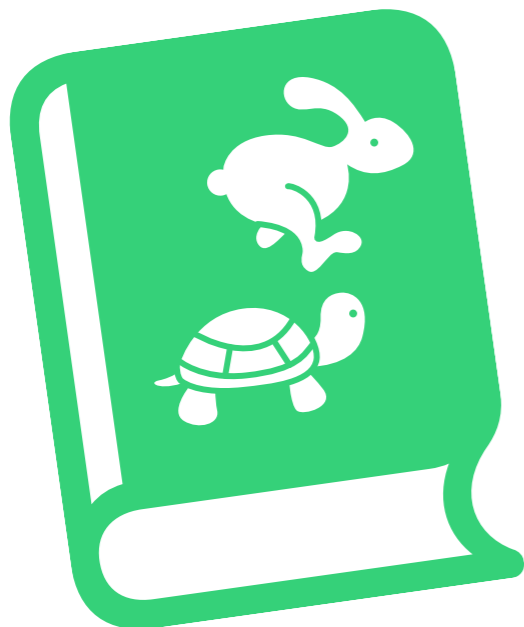
Explorar

Objetivo: descubre cómo los libros siguen una secuencia (desde el principio hasta el final) para que las historias tengan sentido.

Debate:

- Pregunta a los alumnos si los libros siguen una secuencia.
- ¿Qué pasaría si el principio, la mitad y el final de un libro estuvieran desordenados?
- Explora varios ejemplos.

Idea clave: establece la relación con el código, enfatizando lo importante que es dar los comandos de programación en el orden correcto, al igual que los puntos de la trama de una historia.



Descubrir

Objetivo: después de crear imágenes de varios puntos de la trama de una historia, los alumnos podrán colocar las imágenes en orden para recrearla con precisión.

Materiales para los instructores:

- Pizarra
- Rotuladores

Materiales para los alumnos:

- Papel
- Rotuladores o lápices de colores
- Alternativa: dispositivos iPad y una app de dibujo

Indicaciones:

1. Lee una historia que los alumnos conozcan bien. Determinad en grupo los puntos principales de la historia. Lo ideal es pensar entre cuatro y seis puntos.
2. Crea grupos pequeños que tengan el mismo número de alumnos que puntos de la trama; por ejemplo, cuatro puntos de la trama equivalen a cuatro alumnos por grupo.
3. Haz que cada alumno del grupo dibuje uno de los puntos de la trama.
4. Los grupos se turnan para ponerse de pie frente a la clase, con los alumnos desordenados sosteniendo las imágenes de la trama.
5. El público reordena las imágenes, moviéndolas de una en una.
6. Haz una foto de cada grupo cuando los alumnos estén en el orden correcto.

Ampliación de la actividad o alternativa:

Haz que cada grupo de alumnos trabaje en una historia diferente, determinando los puntos de la trama todos juntos antes de hacer los dibujos.

Jugar

Objetivo: los alumnos podrán guiar a Byte a través de una cuadrícula física hasta una gema mediante comandos direccionales.

Preparación: los alumnos trabajarán en grupos de tres. Usa cinta de carrocero para crear una cuadrícula de cuatro por cuatro en el suelo para cada grupo.

Indicaciones:

1. Distribuye los materiales y divide a los alumnos en grupos de tres.
2. Lee cada función y asigna a cada miembro del grupo un rol para el primer juego.
3. Pide a los alumnos que empiecen a jugar, comenzando por el rol de diseñador.
4. Juega tres veces, rotando cada vez las tarjetas de roles.

Roles:

- Diseñador: coloca la gema y la flecha de inicio en la parrilla.
- Programador: con la ayuda de tus compañeros, coloca las tarjetas de comandos en la cuadrícula o junto a ella para dirigir a Byte hasta la gema y recogerla.
- Probador: comenzando con Byte en la flecha, sigue las tarjetas de comandos para mover a Byte por la cuadrícula. Si recogéis la gema, ¡celebradlo! Si no lo conseguís, trabajad en equipo para depurar o corregir el código.

Alternativa:

Si los alumnos están trabajando contigo de forma individual o aprendiendo en casa, pueden jugar a este juego por su cuenta con la actividad alternativa de Keynote, que se pueden descargar.

Materiales para los instructores:

- Cinta de carrocero

Materiales para los alumnos:

- Tarjetas de roles
- Tarjetas de comandos: `moveForward()`, `turnLeft()`, `turnRight()` y `collectGem()`
- Gema
- Byte
- Flecha

↓ [Descarga los materiales](#)

↓ [Descarga la actividad alternativa](#)



Explorar

Objetivo: explora la idea de que la programación puede ser creativa.

Debate:

- Pregunta a los alumnos si alguna vez han aprendido un baile.
- ¿El baile tenía un orden de pasos que seguir?
- ¿Cómo sabían qué tenían que hacer en cada momento?
- ¿Los pasos de baile tienen nombres?
- ¿Los alumnos usan los mismos movimientos en diferentes momentos de un baile o en diferentes coreografías?

Idea clave: ayuda a los alumnos a establecer la conexión de que la programación es creativa y que, como cuando se hace la coreografía de un baile, los programadores pueden crear nuevos comandos y luego combinarlos de maneras diferentes e interesantes.

Descubrir

Objetivo: crea una rutina de baile corta, junto con tarjetas para representar los pasos de baile. Cada tarjeta «Pasos de baile» es como un comando en el área de juegos «Aprende a programar».

Materiales para los alumnos:

- Dispositivos iPad
- App Keynote
- App Cámara
- Espacio para bailar

Indicaciones:

1. Pide a parejas o grupos pequeños de alumnos que creen una breve rutina de baile.
2. Una vez que los alumnos establezcan la rutina, elaborarán tarjetas con los diferentes pasos de baile. Los alumnos deben incluir un dibujo y el nombre del movimiento en cada tarjeta, de modo que sean lo más creativos y graciosos posible.
3. Cada grupo realiza su baile y luego organiza una fiesta de baile con toda la clase.

Alternativa:

Los alumnos pueden usar las tarjetas «Pasos de baile» descargables que aparecen a continuación para crear su baile o pueden usar las tarjetas como ejemplos al crear las suyas propias.

Ampliación de la actividad:

Los alumnos graban un vídeo de su baile para mostrárselo al grupo.



[Descarga las tarjetas «Pasos de baile»](#)



Jugar

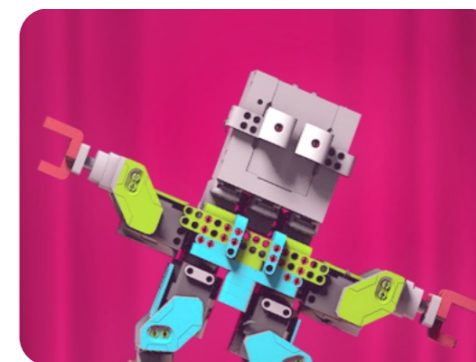
Objetivo: crea una secuencia de pasos para enseñar al robot MeeBot un nuevo baile.

Indicaciones:

1. Proyecta el área de juegos «Danzas MeeBot» en una pantalla. Tendrás que suscribirte al área de juegos si aún no lo has hecho.
2. Introducción:
 - Lee las páginas con toda la clase y haz preguntas si es necesario.
3. Hola, MeeBot:
 - Haz clic o toca Ejecutar código y mira cómo baila el robot.
4. Movimientos básicos:
 - En grupo, en parejas o individualmente en su propio iPad, los alumnos elegirán ocho comandos de la lista de sugerencias y observarán cómo baila el robot.
 - Pide a los alumnos que compartan sus bailes o creen todos juntos varios bailes diferentes.
 - ¡Baila con el robot!

Ampliación de la actividad:

- Pasa a la página siguiente, «Rutina de baile», donde los alumnos pueden añadir movimientos dentro de la función `myDanceRoutine` (): tantos comandos como deseen.



Danzas MeeBot

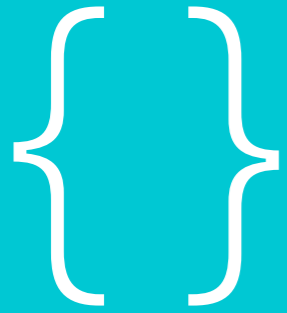
Materiales para los instructores:

- iPad o Mac
- App Swift Playgrounds
- Área de juegos «Danzas MeeBot»
- Proyector o pantalla

Materiales para los alumnos:

- Dispositivos iPad (opcional)





Funciones



Descripción

Lección 1: Gema de papel

- Explorar: debate sobre las instrucciones paso a paso
- Descubrir: actividad «Gema de papel»
- Jugar: «Componer un comportamiento» y «Crear una función nueva»

Lección 2: Festival de música

- Explorar: debate sobre cómo nombrar una función
- Descubrir: actividad «Festival de música»
- Jugar: juego de rompecabezas en el suelo

Lección 3: Mi función relajante

- Explorar: debate sobre cómo resolver problemas de varias formas
- Descubrir: actividad «Mi función relajante»
- Jugar: «Recoger, activar y repetir»

Los alumnos podrán

- Descomponer un problema o una tarea de gran envergadura en pasos más pequeños
- Crear una serie de pasos para resolver un problema o completar una tarea
- Nombrar funciones
- Probar y depurar código

Vocabulario

- **Función:** conjunto de comandos con un nombre que pueden ejecutarse siempre que sea necesario
- **Activar:** encender o apagar

Explorar

Objetivo: explora la idea de empaquetar una serie de comandos y asignarle un nombre.

Debate: decide una rutina diaria en la que se centrará toda la clase. Pide a los alumnos que identifiquen el nombre de su rutina diaria y los pasos que la componen.

Ejemplo: rutina para irse a la cama

- Paso 1: cepillarse los dientes
- Paso 2: ir al baño
- Paso 3: leer
- Paso 4: dar las buenas noches
- Paso 5: apagar las luces

Idea clave: crear un conjunto de instrucciones y darle un nombre es el mismo concepto que crear una función.

Ampliación de la actividad:

pregunta a los alumnos si las instrucciones para alguno de sus pasos podrían ser más específicas. Por ejemplo, ¿cuáles son los pasos específicos para lavarse los dientes?

Descubrir

Objetivo: los alumnos empezarán siguiendo las instrucciones para crear una gema de papel y luego escribirán o dibujarán las instrucciones para hacer otra figura de su elección.

Materiales para los alumnos:

- Papel
- Tijeras
- Lápices
- Dispositivos iPad (opcional)

Indicaciones:

Muestra a los alumnos cómo hacer una gema de papel:

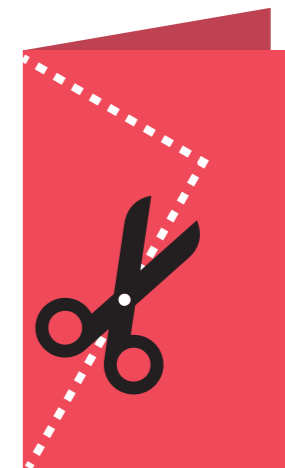
1. Dobla una hoja de papel por la mitad.
2. Dibuja una línea desde la esquina superior del lado doblado hasta 2,5 o 5 centímetros por encima del centro del papel.
3. Dibuja otra línea desde donde termina la primera hasta la esquina inferior del lado doblado.
4. Corta por las líneas que has dibujado.
5. Retira la gema del papel sobrante y despliégala.

Pide a los alumnos que creen sus propias figuras:

1. Divide a los alumnos en grupos pequeños.
2. Pide a los grupos que decidan qué figura hacer.
3. Deja tiempo para que los alumnos practiquen cómo hacer la figura una o dos veces.
4. Pide a los alumnos que escriban o dibujen las instrucciones para hacer la figura y, a continuación, les asignen un nombre, como «Hacer un círculo» o «La letra T».

Alternativa:

Graba un vídeo que muestre cómo hacer sus figuras.



Jugar

Objetivo: trabajando en grupo, los alumnos podrán dividir los pasos necesarios para llevar a Byte hasta la gema.

Indicaciones:

1. Proyecta el área de juegos «Aprende a programar 1» en una pantalla. Ve al capítulo «Funciones» en «Aprende a programar 1».
2. Introducción:
 - Lee las páginas con toda la clase y haz preguntas si es necesario.
3. Componer un comportamiento:
 - Revisa los comandos `moveForward()`, `turnLeft()` y `collectGem()`, teniendo en cuenta que no tienes el comando `turnRight()`.
 - Pide a los alumnos que experimenten formas de dirigir a Byte desde la flecha de inicio hasta la gema para recogerla. A continuación, tendrán que registrar los comandos en la hoja de trabajo o en una hoja de papel aparte.
 - Recopila ideas de la clase y escribe el código en la app Swift Playgrounds para completar el rompecabezas. Haz clic o toca Ejecutar código.
 - Prueba varias ideas diferentes.
 - ¡Celébralo con Byte!
4. Crear una función nueva:
 - Con arreglo a lo que hayan aprendido en la última página del área de juegos «Componer un comportamiento», pide a los alumnos que propongan ideas para crear la función `turnRight()`.
 - Utilizando su función `turnRight()`, pide a los alumnos que experimenten formas de dirigir a Byte desde la flecha de inicio al interruptor cerrado para activarlo.
 - Recopila ideas de la clase y escribe el código en la app Swift Playgrounds para completar el rompecabezas. Haz clic o toca Ejecutar código.
 - Prueba varias ideas diferentes.
 - Celébralo con Byte: ¡era un rompecabezas difícil!



Aprende a programar 1

Materiales para los instructores:

- iPad o Mac
- App Swift Playgrounds
- Área de juegos «Aprende a programar 1»
- Proyector o pantalla

Materiales para los alumnos:

- Hojas de trabajo «Componer un comportamiento» y «Crear una función nueva»
- Lápices
- Papel adicional (opcional)



[Descarga las hojas de trabajo «Aprende a programar»](#)

Explorar

Objetivo: aplica el conocimiento de los comandos y las funciones a las canciones dándoles nombres descriptivos.

Debate: pide a los alumnos que creen una serie de canciones y asignen a cada una un nombre de función descriptivo.

Ejemplo: para la canción «Twinkle, Twinkle, Little Star», la llamada a la función podría ser `singTwinkle()`, pero `singSong1()` no sería un buen nombre porque la primera canción podría cambiar.

Idea clave: asignar nombres a las funciones con nombres descriptivos es importante porque hace que el código sea más fácil de entender para ti y para otras personas.

Descubrir

Objetivo: los alumnos crearán un concierto llamando a diferentes comandos de canción en una función de concierto.

Materiales para los instructores:

- iPad o Mac
- Proyector o pantalla
- Pizarra
- Rotuladores

Indicaciones:

1. Ayuda a los alumnos a crear nombres de funciones para varias canciones, por ejemplo, `singHappyBirthday()`.
2. Elige el orden en grupo para cantar las canciones.
3. Escribe una definición de función para un concierto y completa la función con los comandos de la canción.

Ejemplo:

```
func createConcert() {
    singHappyBirthday()
    singTwinkleTwinkle()
    singMaryHadALittleLamb()
}
createConcert()
```

Alternativa:

Los alumnos cantan en grupos pequeños y cada grupo crea su propia lista de canciones, los nombres de las funciones de las canciones y el orden en el que se cantan. A continuación, cada grupo interpreta sus canciones y graba un vídeo de su concierto.

Jugar

Objetivo: los alumnos resolverán una ecuación simple, colocarán una gema en la respuesta y guiarán a Byte a través de la cuadrícula mediante comandos de dirección.

Preparación: los alumnos trabajarán en grupos de tres. Usa cinta de carrocero para crear una cuadrícula de cuatro por cuatro en el suelo para cada grupo. Coloca la flecha de inicio dentro de un cuadrado y coloca un número dentro de cada uno de los cuadrados restantes.

Indicaciones:

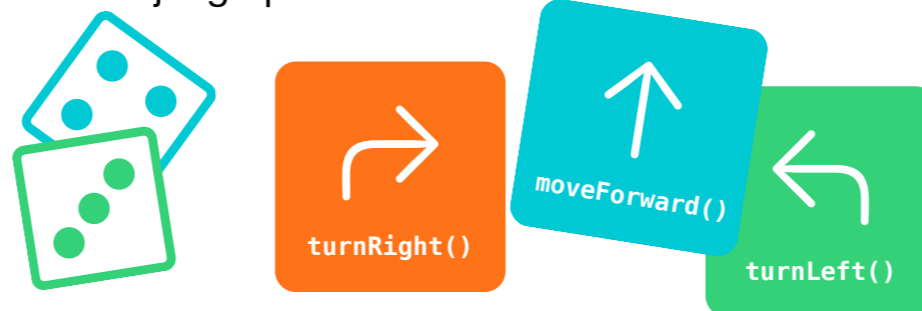
1. Distribuye los materiales y divide a los alumnos en grupos de tres.
2. Lee cada función y asigna a cada miembro del grupo un rol para el primer juego.
3. Pide a los alumnos que empiecen a jugar, comenzando por el rol de diseñador.
4. Juega tres veces, rotando cada vez las tarjetas de roles.

Roles:

- Diseñador: tira dos dados. Con la ayuda de tus compañeros, suma los dos números y coloca la gema en un cuadrado de la cuadrícula que tenga la suma.
- Programador: con la ayuda de tus compañeros, coloca las tarjetas de comandos en la cuadrícula o junto a ella para dirigir a Byte hasta la gema y recogerla.
- Probador: comenzando con Byte en la flecha, sigue las tarjetas de comandos para mover a Byte por la cuadrícula. Si recogéis la gema, ¡celebradlo! Si no lo conseguís, trabajad en equipo para corregir el código.

Alternativa:

Si los alumnos están trabajando contigo de forma individual o aprendiendo en casa, pueden jugar a este juego por su cuenta con la actividad alternativa de Keynote, que se pueden descargar.



Materiales para los instructores:

- Cinta de carrocero
- Un conjunto de números impresos para cada cuadrícula

Materiales para los alumnos:

- Tarjetas de roles
- Tarjetas de comandos: `moveForward()`, `turnLeft()`, `turnRight()` y `collectGem()`
- Gema
- Byte
- Flecha
- Dos dados

- ↓ [Descarga los materiales](#)
- ↓ [Descarga la actividad alternativa](#)

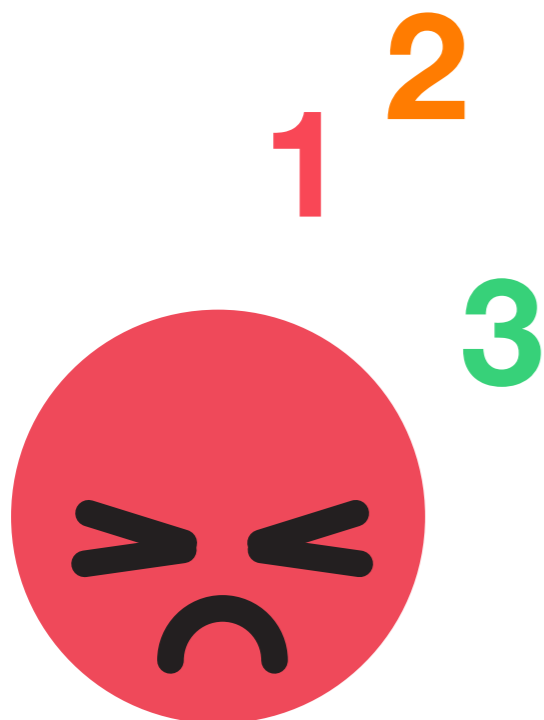


Explorar

Objetivo: los alumnos comprenderán que, por lo general, hay más de una forma de resolver un problema.

Debate: pide a los alumnos que piensen en un problema que hayan tenido y, a continuación, compartan las formas en que lo han solucionado. Pregunta al grupo si alguien podría resolver ese problema de una manera diferente. Explora varios problemas y soluciones diferentes.

Idea clave: ayuda a los alumnos a establecer la relación con el código y aprender que, por lo general, hay más de una forma de resolver un problema de programación.



Descubrir

Objetivo: los alumnos escribirán una función para su técnica relajante y le darán un nombre.

Materiales para los alumnos:

- Hoja de trabajo «Mi función relajante»
- Lápices
- Bolígrafos o lápices de colores

Indicaciones:

Consejo: es mejor que los alumnos trabajen en esta actividad individualmente, si es posible.

1. Pide a los alumnos que piensen en cómo se calman en casa o en la escuela cuando se enfadan. Diles que dividan sus técnicas de relajación en pasos.
2. Reparte la hoja de trabajo «Mi función relajante» y pide a los alumnos que dibujen los pasos de su técnica relajante.
3. Pide a los alumnos que pongan un nombre a su técnica de relajación. Pueden usar notación de camello, por ejemplo, `countToTen()` o simplemente emplear una frase corta, como «Cuenta hasta diez».

Ampliaciones de la actividad:

Actividad sin pantalla: pide a los alumnos que representen su técnica de relajación en grupos pequeños o ante la clase.

Con el iPad: los alumnos graban un vídeo de su técnica relajante para compartirlo con la clase.

↓ [Descarga la hoja de trabajo «Mi función relajante»](#)

Jugar

Objetivo: los alumnos podrán escribir una función compuesta por varios tipos de comandos diferentes y luego usar esa función para completar un rompecabezas.

Indicaciones:

1. Proyecta la página «Recoger, activar y repetir» en el área de juegos «Aprende a programar 1» en una pantalla, señalando la función vacía que los alumnos ayudarán a completar.
2. Recoger, activar y repetir:
 - Revisa los comandos `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` y `toggleSwitch()`.
 - Pide a los alumnos que intenten identificar las partes del rompecabezas que se repiten y que usen sus ideas para completar la función en la app y ponerle un nombre.
 - Pide a los alumnos que inventen un símbolo para la función y registren el símbolo y el nombre de la función en la clave de comandos de la hoja de trabajo.
 - Con el comando adicional, los alumnos experimentan con formas de hacer que Byte recoja todas las gemas y active todos los interruptores. A continuación, registran los comandos en la hoja de trabajo o en una hoja de papel aparte.
 - Recopila ideas de la clase y escribe el código en la app Swift Playgrounds para completar el rompecabezas. Haz clic o toca Ejecutar código.
 - Prueba varias soluciones diferentes.
 - Celébralo con toda la clase: ¡era un rompecabezas difícil!




Aprende a programar 1

Materiales para los instructores:

- iPad o Mac
- App Swift Playgrounds
- Área de juegos «Aprende a programar 1»
- Proyector o pantalla

Materiales para los alumnos:

- Hoja de trabajo «Recoger, activar y repetir»
- Lápices
- Papel adicional (opcional)


[Descarga la hoja de trabajo «Aprende a programar»](#)





Bucles



Descripción

Lección 1: Pétalos que se repiten

- Explorar: debate sobre la relación de la repetición de pasos en el código con la vida real
- Descubrir: actividad «Pétalos que se repiten»
- Jugar: «Utilizar bucles» y «Bucles por los cuatro costados»

Lección 2: Carrera de obstáculos

- Explorar: debate sobre cómo detener puntos en un bucle
- Descubrir: actividad «Carrera de obstáculos»
- Jugar: juego de rompecabezas en el suelo

Lección 3: Patrones de percusión

- Explorar: debate sobre los bucles en la música
- Descubrir: actividad «Patrones de percusión»
- Jugar: «Ida y vuelta» y «Bucles de baile»

Los alumnos podrán

- Identificar un bucle en el código
- Descomponer un problema o una tarea de gran envergadura en pasos más pequeños
- Crear una secuencia de comandos y repetirla mediante un bucle
- Probar y depurar las instrucciones y el código

Vocabulario

- **Bucle:** bloque de código que se repite un número determinado de veces

Explorar

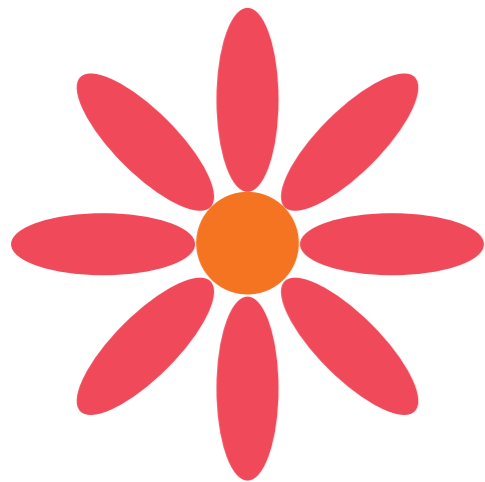
Objetivo: relaciona la idea de los bucles con la vida real.

Debate: explora los momentos en los que los alumnos pueden repetir una tarea o un paso en la vida real.

Ejemplos:

- Caminar
- Montar en bicicleta
- Coser, tejer o hacer ganchillo

Idea clave: los bucles repiten un comando o un conjunto de comandos tantas veces como especifiques.



Descubrir

Objetivo: los alumnos empezarán a explorar el concepto de bucles creando una flor única.

Materiales para los alumnos:

- Hoja de trabajo «Pétalos que se repiten»
- Papel de color
- Lápices
- Tijeras
- Barras de pegamento
- Dados

Indicaciones:

1. Los alumnos dibujan un único pétalo (aproximadamente del largo de la palma de la mano) en una hoja de papel de color y lo recortan. Esta será la plantilla de pétalos de la flor.
2. Cada alumno tira a continuación dos dados, suma los números y rellena el número que falta en el bucle de su hoja de trabajo «Pétalos que se repiten». Este es el número de pétalos que tendrá la flor.
3. Con su plantilla de pétalos, los alumnos trazan el pétalo en papel de color y recortan el número correcto de pétalos para la flor.
4. Con la hoja de trabajo «Pétalos que se repiten», los alumnos montan la flor y pegan las partes en su lugar.



[Descarga la hoja de trabajo «Pétalos que se repiten»](#)



Jugar

Objetivo: los alumnos podrán escribir código dentro de un bucle para recoger todas las gemas.

Indicaciones:

1. Proyecta la página de introducción del capítulo «Bucles For» del área de juegos «Aprende a programar 1» en una pantalla.
2. Introducción:
 - Lee las páginas con toda la clase y haz preguntas si es necesario.
3. Utilizar bucles:
 - Muestra a los alumnos cómo funcionan los portales y revisa los comandos `moveForward()`, `turnLeft()`, `turnRight()` y `collectGem()`.
 - Pide a los alumnos que experimenten formas de dirigir a Byte desde la flecha de inicio hasta las gemas y que observen qué comandos se repiten. A continuación, tendrán que registrar los comandos en la hoja de trabajo o en una hoja de papel aparte.
 - Recopila ideas de la clase y escribe el código en la app Swift Playgrounds para indicar a Byte que recoja la primera gema y se dirija al portal.
 - Pregunta a los alumnos cuántas gemas hay y añade ese número al bucle. Haz clic o toca Ejecutar código.
 - Prueba varias soluciones diferentes.
 - ¡Celebralo con Byte!
4. Bucles por los cuatro costados:
 - Pide a los alumnos que experimenten formas de recoger todas las gemas y que observen qué comandos se repiten.
 - Para añadir un bucle `for`, usa las sugerencias de código en la parte inferior del editor o toca + en la parte superior de la pantalla.
 - Recopila las ideas de la clase y escribe el código en Swift Playgrounds para completar el rompecabezas. Haz clic o toca Ejecutar código.
 - Prueba varias ideas diferentes.
 - ¡Celebralo con Byte!



Aprende a programar 1

Materiales para los instructores:

- iPad o Mac
- App Swift Playgrounds
- Área de juegos «Aprende a programar 1»
- Proyector o pantalla

Materiales para los alumnos:

- Hojas de trabajo «Utilizar bucles» y «Bucles por los cuatro costados»
- Lápices
- Papel adicional (opcional)



[Descarga las hojas de trabajo «Aprende a programar»](#)

Explorar

Objetivo: descubre por qué los bucles siempre necesitan un punto final específico.

Debate: pide a los alumnos que imaginen una noria u otra atracción con la que estén familiarizados. ¿Qué pasaría si el responsable no pulsa el botón para detener el viaje después de cinco vueltas? Pide a los alumnos que proporcionen otros ejemplos de lo que sucedería si no se detiene un bucle.

Idea clave: ayuda a los alumnos a comprender que si no detienen un bucle, se repetirá hasta el infinito.

Descubrir

Objetivo: los alumnos descubrirán cómo funcionan los bucles recorriendo una carrera de obstáculos que hayan diseñado.

Material:

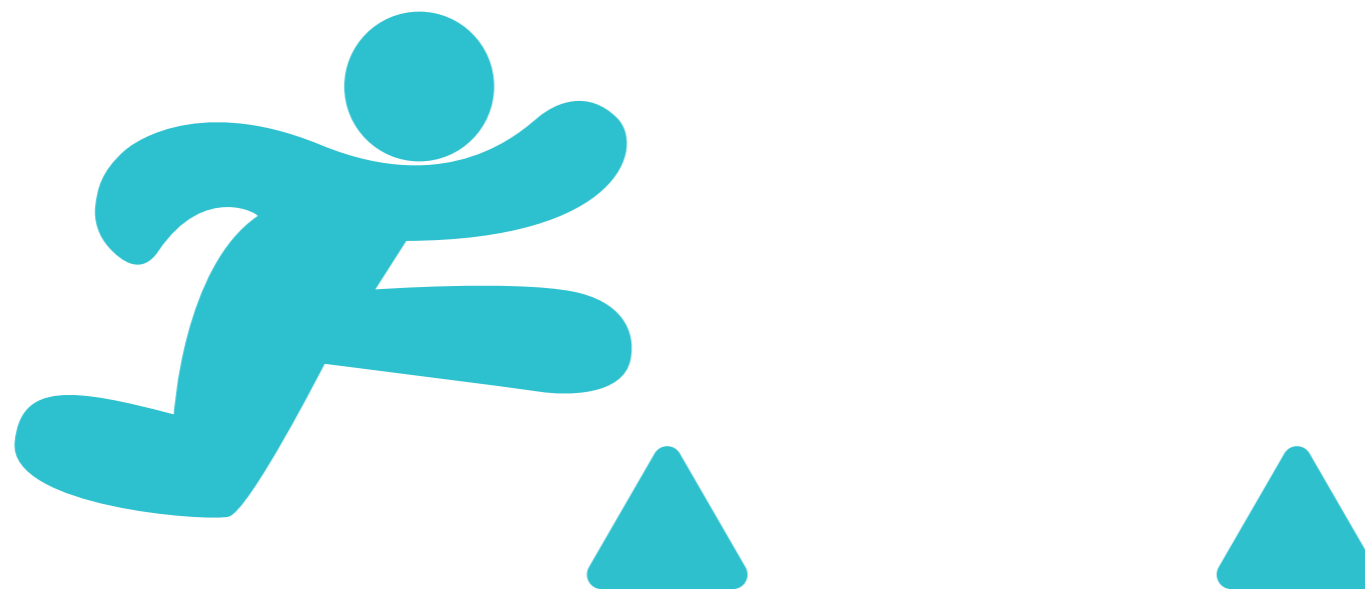
- Espacio para hacer ejercicio físico
- Accesorios para una pista de obstáculos
- Dado

Indicaciones:

1. Crea una breve carrera de obstáculos en el aula o al aire libre.
2. Tira un dado y pide a los alumnos que repitan la pista tantas veces como el número que haya salido en el dado.

Alternativa:

Los alumnos proponen una serie de movimientos: por ejemplo, tocarse los dedos de los pies, saltar, dar una patada. Tira un dado y los alumnos repiten la serie de movimientos el número de veces que indica el dado.



Jugar

Objetivo: los alumnos podrán crear un rompecabezas que tenga un patrón repetitivo y luego resolverlo en grupo.

Preparación: los alumnos trabajarán en grupos de tres. Usa cinta de carrocero para crear una cuadrícula de cuatro por cuatro en el suelo para cada grupo.

Indicaciones:

1. Distribuye los materiales y divide a los alumnos en grupos de tres.
2. Lee cada función y asigna a cada miembro del grupo un rol para el primer juego.
3. Pide a los alumnos que empiecen a jugar, comenzando por el rol de diseñador.
4. Juega tres veces, rotando cada vez las tarjetas de roles.

Roles:

- Diseñador: con la ayuda de tus compañeros, coloca tres gemas en un patrón repetido en la cuadrícula. Coloca la flecha de inicio en la cuadrícula.
- Programador: con la ayuda de tus compañeros, coloca las tarjetas de comandos en la cuadrícula o junto a ella para dirigir a Byte hasta las gemas y recogerlas. Usa las tarjetas de bucles para indicar al probador cuántas veces debe repetir los comandos.
- Probador: comenzando con Byte en la flecha, sigue las tarjetas de comandos para mover a Byte por la cuadrícula. Si recogéis todas las gemas, ¡celebradlo! Si no lo conseguís, trabajad en equipo para corregir el código.

Alternativa:

Si los alumnos están trabajando contigo de forma individual o aprendiendo en casa, pueden jugar a este juego por su cuenta con la actividad alternativa de Keynote, que se pueden descargar.

Materiales para los instructores:

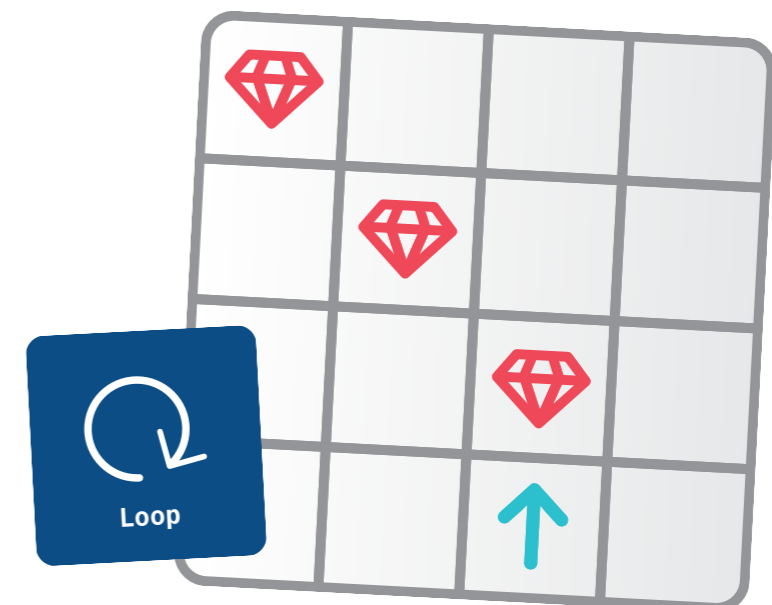
- Cinta de carrocero

Materiales para los alumnos:

- Tarjetas de roles
- Tarjetas de comandos: `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` y `Loop`
- Gemas
- Byte
- Flecha

↓ [Descarga los materiales](#)

↓ [Descarga la actividad alternativa](#)



Explorar

Objetivo: explora los patrones que se repiten en la música.

Debate: pide a los alumnos que compartan los instrumentos que han tocado o las canciones que han cantado. Pregúntales si alguna vez repiten un ritmo o un estribillo cuando están tocando o cantando. ¿Pueden pensar en otras partes de una canción o música que se repitan?

Idea clave: refuerza la idea de que los bucles constan de dos partes:

- Los comandos
- El número de veces que se repiten



Descubrir

Objetivo: los alumnos podrán repetir un patrón de percusión, estableciendo una conexión entre el código en bucle y un ejemplo físico de la vida real.

Material:

- Algo sobre lo que tamborilear, como el suelo, los muslos o los libros
- Espacio para sentarse en círculo

Indicaciones:

1. Pide a los alumnos que se sienten en círculo.
2. Dile a los alumnos que repitan el ritmo que has creado tantas veces como dedos levantes. Por ejemplo, si levantas cuatro dedos, los alumnos deben repetir el ritmo de percusión cuatro veces y luego detenerse.
3. Turnaos para dar la vuelta al círculo o dividíos en grupos pequeños para que cada alumno tenga la oportunidad de ser el percusionista principal.

Ampliación de la actividad:

Pide a los alumnos que creen percusiones.

Jugar

Objetivo: los alumnos llamarán a varios comandos diferentes dentro de un bucle y determinarán cuántas veces se debe llamar al bucle.

Indicaciones:

1. Proyecta el área de juegos «Aprende a programar 1» en una pantalla. Ve a la tercera página, «Ida y vuelta», del capítulo «Bucles for».
2. Ida y vuelta:
 - Revisa los comandos `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` y `toggleSwitch()`.
 - Pide a los alumnos que experimenten formas de dirigir a Byte desde la flecha de inicio a cada interruptor cerrado para activarlo.
 - Para añadir un bucle `for`, usa las sugerencias de código en la parte inferior del editor o toca + en la parte superior de la pantalla.
 - Recopila ideas de la clase y escribe el código en la app Swift Playgrounds para completar el rompecabezas. Haz clic o toca Ejecutar código.
 - Prueba varias ideas diferentes.
 - ¡Celébralo con Byte!
3. Sal de «Aprende a programar 1» y abre el área de juegos «Danzas MeeBot» para ir a la página «Bucles de baile». (Esta página del área de juegos no tiene una hoja de trabajo para los alumnos).
4. Bucles de baile:
 - Pide a los alumnos que trabajen en grupo, en parejas o individualmente en su propio iPad para completar el bucle y ver al robot bailar.
 - Pide a los alumnos que compartan sus bailes o que creen todos juntos algunos diferentes.
 - ¡Baila con el robot!



Aprende a programar 1




Danzas MeeBot

Materiales para los instructores:

- iPad o Mac
- App Swift Playgrounds
- Área de juegos «Aprende a programar 1»
- Área de juegos «Danzas MeeBot»
- Proyector o pantalla

Materiales para los alumnos:

- Hoja de trabajo «Ida y vuelta»
- Lápices
- Dispositivos iPad (opcional)
- Papel adicional (opcional)


[Descarga la hoja de trabajo «Aprende a programar»](#)



Variables



Descripción

Lección 1: Hundirse o flotar

- Explorar: debate sobre la actualización de una variable
- Descubrir: actividad «Hundirse o flotar»
- Jugar: «Llevar el recuento» y «Juego de muestra»

Lección 2: Juego de palabras

- Explorar: debate sobre los tipos de respuestas a las preguntas
- Descubrir: actividad «Juego de palabras»
- Jugar: juego de rompecabezas en el suelo

Lección 3: Todo sobre mí

- Explorar: debate sobre cómo responder preguntas con listas
- Descubrir: actividad «Todo sobre mí»
- Jugar: «Utilizar un bucle»

Los alumnos podrán

- Asociar un nombre de variable a un valor determinado
- Cambiar el valor asignado a una variable
- Comprender los diferentes tipos de Swift que puedes asignar a una variable, incluidos verdadero/falso (booleanos), números (Ints), palabras (cadenas), colores (literales de color) e imágenes (literales de imagen)
- Probar y depurar las instrucciones y el código

Vocabulario

- **Variable:** contenedor que tiene un nombre y almacena un valor que se puede cambiar
- **Datos:** información
- **Booleano:** un tipo que tiene el valor verdadero o falso

Explorar

Objetivo: explora el concepto de variables contando objetos y actualizando el número de variables.

Materiales para los instructores:

- Pizarra
- Rotulador
- Borrador
- Contenedor
- Cinco lápices (o cinco unidades del mismo objeto)

Indicaciones:

1. Comienza escribiendo una sentencia de variable en la pizarra para realizar un seguimiento de los objetos.
 - Ejemplo: `var numberOfPencils = 0`
2. Sostén un contenedor vacío e indica a los alumnos que el contenedor representa tu variable, `numberOfPencils`.
3. Añade un lápiz al contenedor y pregunta a los alumnos cuál es ahora el recuento de la variable. Cuando respondan correctamente, borra el *0* y escribe *1*.
4. Continúa hasta que hayas añadido todos los lápices y el código sea: `var numberOfPencils = 5`.
5. A continuación, empieza a sacar lápices del contenedor y actualiza la variable a medida que los vayas quitando.

Idea clave: ayuda a los alumnos a comprender que las variables almacenan cierta información. En este caso, la información es un número que indica cuántos lápices hay en el contenedor.

Descubrir

Objetivo: con los objetos encontrados, los alumnos realizarán experimentos para determinar si los elementos se hunden o flotan y, a continuación, registrarán los datos utilizando imágenes (literales de imagen) y valores de verdadero/falso (booleanos).

Materiales para los alumnos:

- Dispositivos iPad
- App Keynote
- Hoja de trabajo «Hundirse o flotar»
- Cubo de agua
- Varios objetos para probar

Indicaciones:

1. Divide a los alumnos en grupos pequeños.
2. Diles que cojan varios elementos para probar.
3. Respecto de cada elemento, pide a los alumnos que:
 - Hagan una foto del elemento y la añadan a la hoja de trabajo.
 - Prueben el elemento en el agua.
 - Registren los resultados en la hoja de trabajo haciendo un círculo alrededor de verdadero o falso.



[Descarga la hoja de trabajo «Hundirse o flotar»](#)

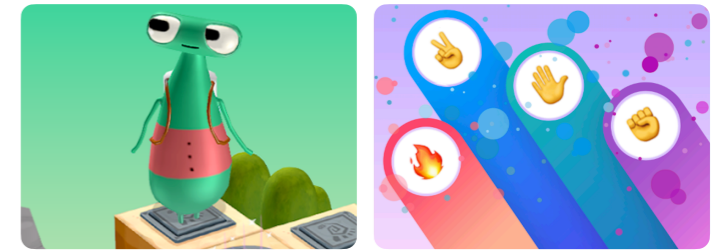
Jugar

Objetivo: los alumnos podrán crear y actualizar variables en dos contextos de programación diferentes.

Indicaciones:

1. Proyecta el área de juegos «Aprende a programar 2» en una pantalla. Ve al capítulo «Variables».
2. Introducción:
 - Lee las páginas con toda la clase y haz preguntas si es necesario.
3. Llevar el recuento:
 - Pide a los alumnos que experimenten formas de dirigir a Hopper desde la flecha de inicio hasta la gema para recogerla. A continuación, tendrán que registrar los comandos en la hoja de trabajo o en una hoja de papel aparte.
 - Recopila ideas de la clase y escribe el código en la app Swift Playgrounds para completar el rompecabezas. Haz clic o toca Ejecutar código.
 - Prueba varias ideas diferentes.
 - ¡Celebralo con Hopper!
4. Sal de «Aprende a programar 2» y pasa a la última página del área de juegos «Piedra, papel o tijera», llamada «Juego de muestra». (Esta página del área de juegos no tiene una hoja de trabajo para los alumnos).
5. Juego de muestra:
 - Haz clic o toca Ejecutar código para jugar antes de cambiar nada.
 - Decide en grupo qué partes del juego queréis personalizar. Algunas cosas divertidas que puedes cambiar son `game.roundsToWin`, `game.challenger.emoji`, `game.addOpponent` y `game.roundPrize`.
 - Juega varias veces, cambiando algo diferente cada vez.

Ampliación de la actividad: muchas variables se establecen en el archivo `Game.swift`. Si los alumnos tienen curiosidad por saber por qué algunas variables no tienen `var` delante de ellas, abre el archivo `Game.swift` para mostrarles dónde se crearon las propiedades del juego.



Aprende a programar 2

Piedra, papel o tijera

Materiales para los instructores:

- iPad o Mac
- App Swift Playgrounds
- Área de juegos «Aprende a programar 2»
- Área de juegos «Piedra, papel o tijera»
- Proyector o pantalla

Materiales para los alumnos:

- Hoja de trabajo «Llevar el recuento»
- Lápices
- Papel adicional (opcional)



[Descarga la hoja de trabajo «Aprende a programar»](#)

Explorar

Objetivo: explora varios tipos de respuestas en el mundo real y relaciónalos con varios tipos de Swift, incluidos sí/no o verdadero/falso (booleanos), números (Ints), palabras (cadenas), colores (literales de color) e imágenes (literales de imagen).

Materiales para los instructores:

- Pizarra
- Rotuladores

Debate: pide a la clase que proponga algunas preguntas que requieran diferentes tipos de respuestas y escríbelas en la pizarra.

Ejemplos:

- ¿De qué color son tus ojos? → color
- ¿Tienes mascota? → sí/no
- ¿Tienes hermanos? → sí/no
- ¿Cuántos años tienes? → número
- ¿Cómo te llamas? → palabra

Idea clave: explica que las variables también tienen diferentes tipos, incluidos números, palabras, colores, imágenes y respuestas sí o no. Dependiendo de cómo crees una variable, tendrás que mantener el mismo tipo, incluso si actualizas la variable a algo nuevo. Por ejemplo, `var myAge = 8` puede cambiar a 9, pero no puede cambiar a «nueve».

Descubrir

Objetivo: los alumnos podrán completar un juego de palabras introduciendo el tipo de respuesta correcto.

Materiales para los alumnos:

- Hojas de trabajo «Juego de palabras»
- Lápices
- Lápices de colores

Indicaciones:

Pide a los alumnos que trabajen en uno o más juegos de palabras en grupos pequeños. Lo ideal es que cada grupo tenga al menos un lector o una persona de apoyo. Si ningún alumno sabe leer, haz algunos juegos en grupo.

Ampliación de la actividad: si los alumnos son capaces, pídeles que creen un juego de palabras para que lo complete un compañero. Anímalos a usar números, palabras, colores, imágenes y respuestas sí o no para los espacios en blanco.



[Descarga las hojas de trabajo «Juego de palabras»](#)

Jugar

Objetivo: los alumnos podrán guiar a Byte para que recoja varias gemas, añada cada gema a un contenedor y actualice una variable.

Preparación: los alumnos trabajarán en grupos de tres. Usa cinta de carrocero para crear una cuadrícula de cuatro por cuatro en el suelo para cada grupo.

Indicaciones:

1. Distribuye los materiales y divide a los alumnos en grupos de tres.
2. Lee cada función y asigna a cada miembro del grupo un rol para el primer juego.
3. Pide a los alumnos que empiecen a jugar, comenzando por el rol de diseñador.
4. Juega tres veces, rotando cada vez las tarjetas de roles.

Roles:

- Diseñador: coloca varias gemas y la flecha de inicio en la cuadrícula.
- Programador: con la ayuda de tus compañeros, coloca las tarjetas de comandos en la cuadrícula o junto a ella para dirigir a Byte hasta las gemas y recogerlas.
- Probador: empezando con Byte en la flecha, sigue los comandos para mover a Byte por la cuadrícula y añadir las gemas al contenedor a medida que las recoges. Si recoges todas las gemas, actualiza la variable `numberOfGems` en el contenedor y celébralo. Si no las recoges todas, trabaja en equipo para corregir el código.

Alternativa:

Si los alumnos están trabajando contigo de forma individual o aprendiendo en casa, pueden jugar a este juego por su cuenta con la actividad alternativa de Keynote, que se pueden descargar.

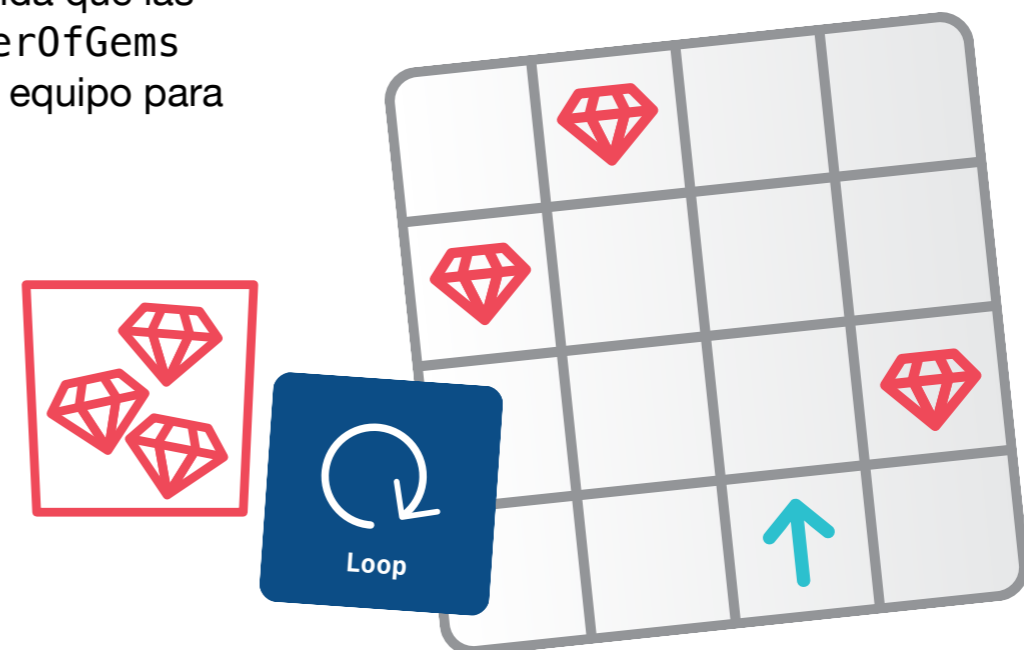
Materiales para los instructores:

- Cinta de carrocero

Materiales para los alumnos:

- Tarjetas de roles
- Tarjetas de comandos: `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` y `Loop`
- Gemas
- Byte
- Flecha
- Contenedor etiquetado: `var numberOfGems = _____`
- Bolígrafo

- ↓ [Descarga los materiales](#)
- ↓ [Descarga la actividad alternativa](#)












Explorar

Objetivo: descubre cómo usar las listas, o *vectores*, al crear variables.

Debate: ¿qué pasaría si en una hoja de trabajo se preguntara a los alumnos el nombre de sus hermanos y tuvieran más de uno? Recopila ideas de la clase. Si sugieren que hagas una lista, diles que justo eso es lo que hacen los programadores. Cuando una variable tiene más de una respuesta, los alumnos deben crear una lista.

Pide a los alumnos que planteen preguntas que podrían tener varias respuestas.

Ejemplos:

- Nombres de amigos —> Rose, Sam, Joy
- Edades de los alumnos —> 7, 8, 7, 8, 7, 8, 9, 7, 8, 9, 8
- Colores favoritos —> , , , , 
- Animales favoritos —> , , , 

Idea clave: las listas que los alumnos crean en el código son como las listas de una frase.

Descubrir

Objetivo: los alumnos podrán introducir variables para describir cosas sobre ellos mismos y un compañero. Los alumnos pueden tener la oportunidad de usar un vector como tipo de variable.

Materiales para los alumnos:

- Hojas de trabajo «Todo sobre mí» y «Todo sobre ti»
- Lápices
- Lápices de colores

Indicaciones:

1. Pide a los alumnos que completen la hoja de trabajo «Todo sobre mí».
 - Si los alumnos tienen más de un hermano o una mascota, pídeles que hagan una lista de elementos separados por comas.
2. Empareja a los alumnos para completar la hoja de trabajo «Todo sobre ti».

Alternativa: los alumnos pueden usar el iPad y Keynote para completar la hoja de trabajo, hacer fotos para las respuestas con imágenes y colorear los literales de color con las opciones de formato.



[Descarga las hojas de trabajo «Todo sobre»](#)

Jugar

Objetivo: los alumnos podrán identificar una variable en el código y explorar formas de usar vectores con bucles.

Indicaciones:

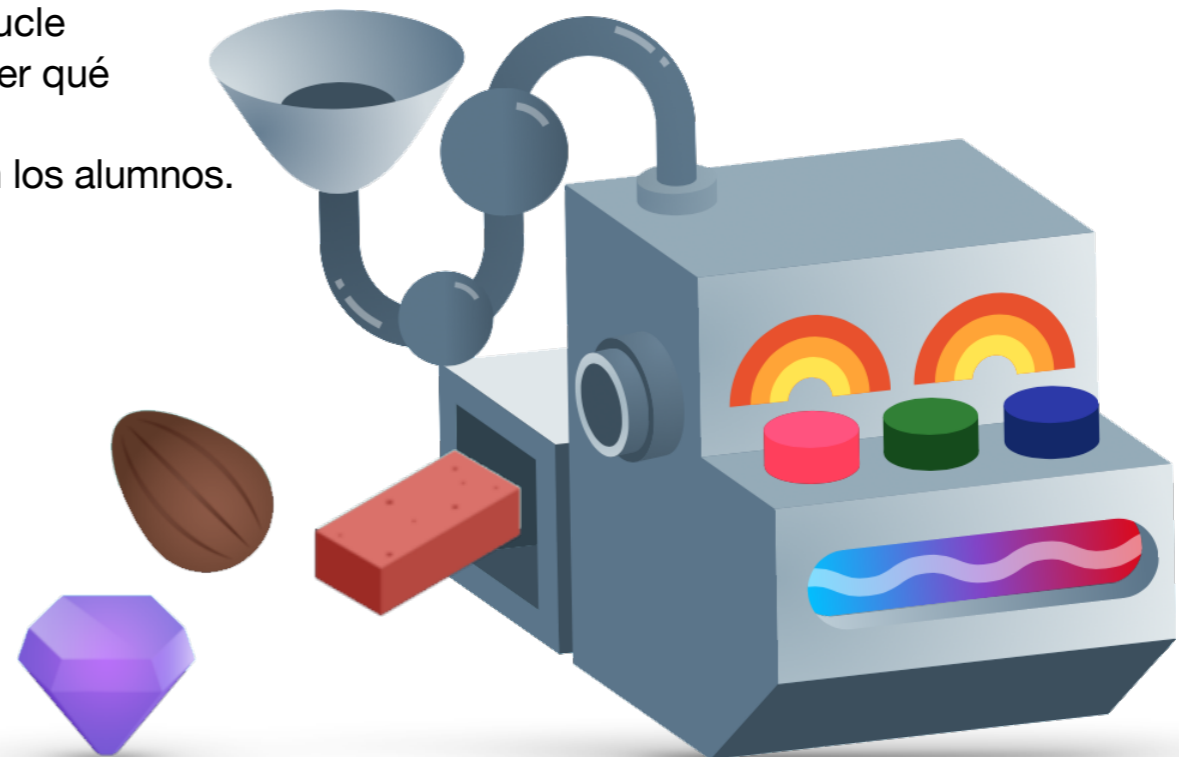
1. Proyecta el área de juegos «Máquina de código» en una pantalla.
2. Introducción:
 - Lee las páginas con toda la clase y haz preguntas si es necesario.
 - Opcional: reproduce las dos primeras páginas, «Explorar la máquina» y «Combinar con colores».
3. Utilizar un bucle:
 - En esta página, los alumnos combinarán sus conocimientos sobre bucles con variables.
 - Comprueba si los alumnos pueden identificar la variable del código que usa un vector.
 - Haz clic o toca Ejecutar código para ver qué crea la máquina.
 - Continúa con el segundo paso de las instrucciones y actualiza el código para que incluya una segunda variable, elementos y un bucle anidado. Haz clic o toca Ejecutar código de nuevo para ver qué crea la máquina.
 - Nota: Prueba esta página antes de realizar la actividad con los alumnos.



Máquina de código

Materiales para los instructores:

- iPad o Mac
- App Swift Playgrounds
- Área de juegos «Máquina de código»
- Proyector o pantalla



Diseño de apps



Explorar

Objetivo: explora apps conocidas en varios dispositivos.

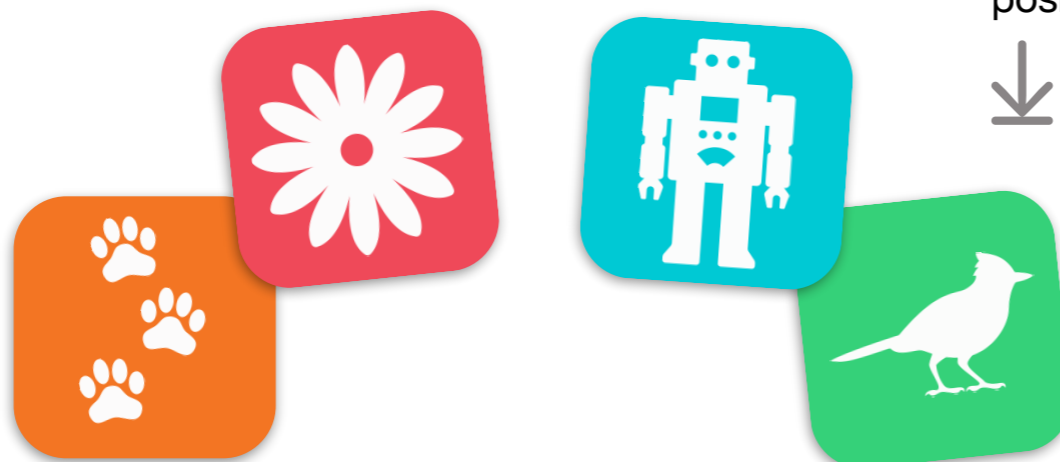
Indicaciones: inicia un debate sobre las apps que los alumnos usan en el iPad en casa o en el centro educativo. A continuación, habla de las apps que ellos o sus padres o tutores usan en los dispositivos de casa.

Idea clave: refuerza la idea de que hay apps no solo en los teléfonos, sino también en los relojes, las tabletas, los ordenadores e incluso en los televisores.

Ampliación de la actividad: profundiza en algunos ejemplos de apps y pregunta a los alumnos para quién está diseñada, para qué sirve y por qué creen que se ha creado.

Ejemplo:

- App: Swift Playgrounds
- A quién va dirigida: personas que quieren aprender más sobre Swift
- Qué hace: ayuda a las personas a aprender a programar mediante rompecabezas y lecciones
- Por qué se creó: para enseñar a programar a personas con pocos o ningún conocimiento de programación



Descubrir

Objetivo: prepara a los alumnos para que diseñen sus propias apps analizando una que ya conozcan.

Materiales para los alumnos:

- Dispositivos iPad
- Hoja de trabajo «¿Qué es una app?»
- Lápices
- Bolígrafos o lápices de colores

Indicaciones:

1. Divide a los alumnos en grupos pequeños o pídeles que trabajen individualmente.
2. Pide a los alumnos que elijan una app para el iPad.
3. Diles que usen la hoja de trabajo «¿Qué es una app?» para guiarlos en su exploración de la app.
4. Invita a los alumnos a compartir sus descubrimientos sobre la app, ya sea con todo el grupo o con algunos compañeros.

Consejo para los instructores: cuanto más jóvenes sean los alumnos, más ayuda necesitarán para completar esta hoja de trabajo. Para alumnos de 5 o 6 años, considera la posibilidad de hacer dos o tres apps en grupo.



[Descarga la hoja de trabajo «¿Qué es una app?»](#)

Jugar

Objetivo: los alumnos diseñan sus propias apps.

Materiales para los alumnos:

- Hoja de trabajo «Mi diseño de app»
- Plantillas de dispositivos
- Papel adicional
- Lápices
- Bolígrafos o lápices de colores

Indicaciones:

1. Divide a los alumnos en grupos pequeños o pídeles que trabajen individualmente.
2. Explica a los alumnos la hoja de trabajo «Mi diseño de app» para guiarlos en el proceso inicial de diseño de una app.
3. Pide a los alumnos que creen prototipos de las páginas de sus apps utilizando papel adicional o las plantillas de dispositivos.
4. Indica a los alumnos que creen una versión final de los prototipos de sus apps utilizando las plantillas de dispositivos.
5. Invita a cada alumno o grupo de alumnos a presentar sus ideas a todo el grupo.

↓ [Descarga la hoja de trabajo «Mi diseño de app»](#)

↓ [Descarga las plantillas de dispositivos](#)



Recursos para los instructores



Glosario

- **Booleano:** un tipo que tiene el valor verdadero o falso
- **Error:** un error de código
- **Comando:** código que indica a una aplicación que realice una acción específica
- **Datos:** información
- **Depurar:** encontrar y corregir errores de código
- **Función:** conjunto de comandos con un nombre que pueden ejecutarse siempre que sea necesario
- **Bucle:** bloque de código que se repite un número determinado de veces
- **Modificar:** cambiar
- **Secuencia:** el orden en el que suceden las cosas
- **Paso:** una acción en un proceso más amplio
- **Activar:** encender o apagar
- **Variable:** contenedor que tiene un nombre y almacena un valor que se puede cambiar

Respuestas de ejemplo

En las páginas siguientes encontrarás una posible solución para cada rompecabezas de Swift Playgrounds, pero se pueden resolver de más de una forma. Anima a los alumnos a probar diferentes formas de dirigir a Byte u otros personajes.

Celebra todos los tipos de programación y objetivos que puedan tener los alumnos. Es posible que algunos alumnos quieran explorar todo el espacio del rompecabezas además de recoger las gemas, mientras que otros quizá deseen girar tantas veces como sea posible antes de recogerlas. Recuerda, ¡programar debe ser divertido!



Aprende a programar 1

Capítulo de comandos

Dar órdenes con comandos

```
moveForward()
moveForward()
moveForward()
collectGem()
```

Capítulo de comandos

Añadir un comando nuevo

```
moveForward()
moveForward()
turnLeft()
moveForward()
moveForward()
collectGem()
```

Capítulo de funciones

Componer un comportamiento

```
moveForward()
moveForward()
moveForward()
turnLeft()
turnLeft()
turnLeft()
moveForward()
moveForward()
moveForward()
collectGem()
```

Capítulo de funciones

Crear una función nueva

```
func turnRight() {
    turnLeft()
    turnLeft()
    turnLeft()
}

moveForward()
turnLeft()
moveForward()
turnRight()
moveForward()
turnRight()
moveForward()
turnLeft()
moveForward()
toggleSwitch()
```




Aprende a programar 1

Capítulo de funciones

Recoger, activar y repetir

```
func collectToggle() {
  moveForward()
  collectGem()
  moveForward()
  toggleSwitch()
  moveForward()
}
```

```
collectToggle()
turnLeft()
collectToggle()
moveForward()
turnLeft()
collectToggle()
turnLeft()
collectToggle()
```

Capítulo de bucles

Utilizar bucles

```
for i in 1 ... 5 {
  moveForward()
  moveForward()
  collectGem()
  moveForward()
}
```

Capítulo de bucles

Bucles por los cuatro costados

```
for i in 1 ... 4 {
  moveForward()
  collectGem()
  moveForward()
  moveForward()
  moveForward()
  turnRight()
}
```

Capítulo de bucles

Ida y vuelta

```
for i in 1 ... 4 {
  moveForward()
  moveForward()
  toggleSwitch()
  turnLeft()
  turnLeft()
  moveForward()
  moveForward()
  turnLeft()
}
```

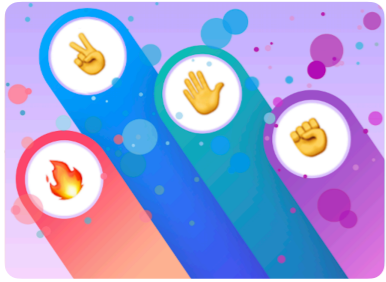


Aprende a programar 2

Capítulo de variables

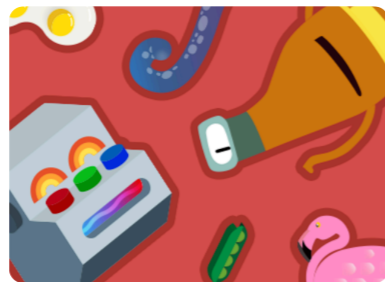
Llevar el recuento

```
var gemCounter = 0
moveForward()
moveForward()
collectGem()
gemCounter += 1
```



Piedra, papel o tijera
Juego de muestra

No hay ningún ejemplo de solución para esta página porque el juego es totalmente personalizable: puedes jugarlo de cualquier manera que te guste.

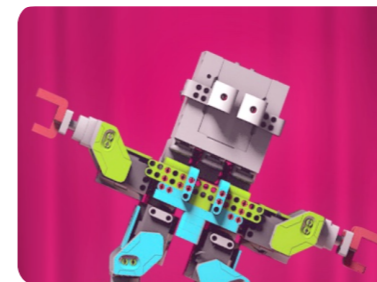


Máquina de código
Utilizar un bucle

```
var colors = [Light.red, Light.green,
Light.blue]

var items = [Item.metal, Item.stone,
Item.cloth, Item.dirt, Item.DNA,
Item.spring, Item.wire, Item.egg,
Item.tree, Item.gear, Item.seed,
Item.crystal, Item.mushroom,
Item.unidentifiedLifeForm]

for item in items {
  setItemA(item)
  setItemB(.dirt)
  switchLightOn(.green)
  forgeItems()
}
```



Danzas MeeBot
Movimientos básicos

```
bendAndTwist()
happy()
moveBackward()
shake()
skip()
split()
swagger()
twist()
```

Bucles de baile

```
for i in 1 ... 5 {
  bend()
  bend(beats: 2)
  bendAndTwist()
  moveBackward(beats: 9)
}
```

