

Programación para todos Estudiantes de nivel inicial

Guía para profesores



Índice

Introducción

Comandos

- Lección 1: Rutinas diarias
- Lección 2: Orden de la historia
- Lección 3: Pasos de baile

Funciones

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

Ciclos

- Lección 1: Pétalos repetidos
- Lección 2: Carrera de obstáculos
- Lección 3: Patrones de percusión

Variables

- Lección 1: ¿Se hunde o flota?
- Lección 2: Juego de palabras
- Lección 3: Todo sobre mí

Diseño de apps

Recursos para el facilitador



Programación para todos:
Estudiantes de nivel inicial está diseñado para ayudar a los educadores y las familias a presentar la programación en los primeros grados, cuando los estudiantes están desarrollando por primera vez sus habilidades de pensamiento computacional. A través de estas lecciones, los estudiantes de jardín de infantes a tercer grado explorarán, descubrirán y jugarán para crear una base de los conceptos básicos de programación.



Diseño del material didáctico

Esta guía se divide en cuatro módulos y finaliza con un proyecto 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

Presenta y analiza el concepto de programación.

Descubrir

Haz que los estudiantes se familiaricen con el concepto mediante actividades creativas.

~25
minutos

Reproducir

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

~25
minutos

Día 2: Relacionar el aprendizaje con el código

Alcance y secuencia

Los cuatro módulos de esta guía están diseñados para usarlos desde el jardín de infantes hasta el tercer grado y se pueden realizar en cualquier orden. Recomendamos que uses el módulo Diseño de apps en cualquier momento o incluso varias veces al año a medida que los estudiantes comprendan mejor el código y las apps.

Ejemplo:

Grado	Módulo	Proyecto final	Tiempo total aproximado
Jardín de infantes	Comandos	Diseño de apps	4 horas
Primero	Funciones	Diseño de apps	4 horas
Segundo	Ciclos	Diseño de apps	4 horas
Tercero	Variables	Diseño de apps	4 horas

Sigue aprendiendo

Para enseñar a estudiantes de cuarto a octavo grado, los planes de estudio de Rompecabezas y Aventuras de Programación para todos, junto con el Diario de diseño de apps y la Guía de exhibición de apps, ofrecen más de 90 horas de aprendizaje. Obtén más información en la [Guía del plan de estudios de Programación para todos](#).



Carpeta de trabajos de los estudiantes (opcional)

Durante todos estos módulos, recopila elementos de las actividades para crear carpetas de trabajos con tus estudiantes.



Módulo	Lección	Elementos sugeridos
Comandos	Rutinas diarias	<ul style="list-style-type: none"> • Hoja de trabajo Crea comandos • Hoja de trabajo Agrega un nuevo comando
	Orden de la historia	<ul style="list-style-type: none"> • Imagen de puntos de la trama de Orden de la historia • Imagen grupal de Orden de la historia
	Pasos de baile	<ul style="list-style-type: none"> • Tarjetas de Pasos de baile • Video de Pasos de baile (opcional)
Funciones	Gema de papel	<ul style="list-style-type: none"> • Forma de Gema de papel • Hoja de trabajo Nuevos comportamientos • Hoja de trabajo Crea una nueva función
	Festival de canciones	<ul style="list-style-type: none"> • Video del concierto o función escrita de Festival de canciones
	Mi función relajante	<ul style="list-style-type: none"> • Video o dibujo de la actividad Mi función relajante • Hoja de trabajo Recolecta, activa y repite
Ciclos	Pétalos repetidos	<ul style="list-style-type: none"> • Pétalos repetidos • Hoja de trabajo Ciclos • Hoja de trabajo Ciclos por todos lados
	Carrera de obstáculos	<ul style="list-style-type: none"> • Video o imágenes de la carrera de obstáculos (opcional)
	Patrones de percusión	<ul style="list-style-type: none"> • Hoja de trabajo Hacia afuera y hacia adentro • Video o imágenes de la percusión (opcional)
Variables	¿Se hunde o flota?	<ul style="list-style-type: none"> • ¿Se hunde o flota? • Hoja de trabajo Hacer un seguimiento
	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 una app		<ul style="list-style-type: none"> • ¿Qué es una app? • Diseño de mi app • Prototipo de diseño de apps

Primeros pasos en Swift Playgrounds en iPad o Mac



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

Los módulos de esta guía usan distintas combinaciones de playgrounds. Esto es lo que necesitarás para cada módulo:

Módulo	Playgrounds	Cómo hacer descargas en Swift Playgrounds
Comandos	 Aprender a programar 1  El baile de MeeBot	El baile de MeeBot: En la sección “De otros editores” en la pantalla Más playgrounds, toca UBTech Jimu Robots y, luego, toca Suscribirse. Toca Obtener para descargar el playground El baile de MeeBot 2.0.
Funciones	 Aprender a programar 1	
Ciclos	 Aprender a programar 1  El baile de MeeBot	El baile de MeeBot: En la sección “De otros editores” en la pantalla Más playgrounds, toca UBTech Jimu Robots y, luego, toca Suscribirse. Toca Obtener para descargar el playground El baile de MeeBot 2.0.
Variables	 Aprender a programar 2  Piedra, papel o tijeras  Máquina de código	Piedra, papel o tijeras y Máquina de código: En la sección “Desafíos” de la pantalla Más playgrounds, toca Obtener para descargar los playgrounds.
Diseño de una app		

Revisa los requerimientos mínimos de Swift Playgrounds en el [App Store](#). Visita la página de [Soporte técnico de Apple](#) si necesitas ayuda con Swift Playgrounds.

Consejos para el facilitador

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

Actividades de Explorar y descubrir:

- Simplifica cualquier sintaxis o mayúsculas especiales 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 de juego:

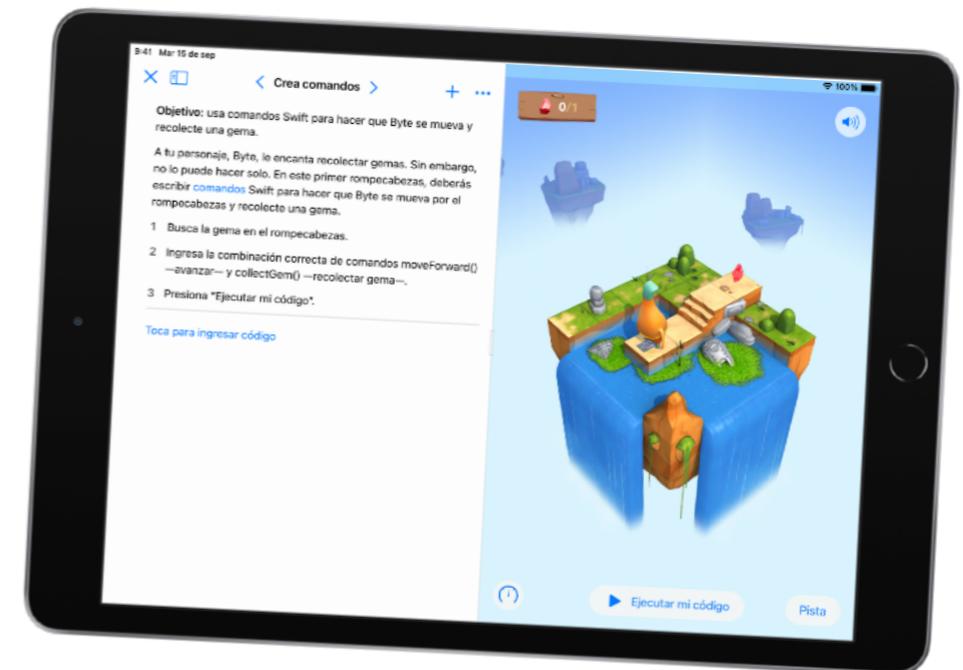
- Para hacer que la app Swift Playgrounds sea aún más simple para tus pequeños estudiantes, sigue las instrucciones en los planes de lecciones. Entre ellos, se incluyen los siguientes:
 - Leer las introducciones juntos como grupo
 - Ofrecer instrucciones simplificadas a los estudiantes para las hojas de trabajo adjuntas para que puedan encontrar sus propias soluciones
 - Usar un iPad o Mac facilitador para resolver los rompecabezas en la app
- `let` y `var`: esta guía no abarca la palabra clave `let`. Para evitar confusiones en Swift Playgrounds, cambia la palabra clave `let` por `var` antes de mostrar las páginas a los estudiantes. En los playgrounds que recomendamos, las dos palabras clave son intercambiables.
 - `let` = la variable no cambia
 - `var` = la variable cambia

Extensiones:

- Amplía las actividades de juego en el suelo para que incluyan aritmética, alfabetización, palabras reconocibles a simple vista, ortografía y más. Prueba la actividad de juego en el suelo en el módulo Funciones como inspiración.
- Personaliza las actividades de juego en el suelo al hacer que los estudiantes elaboren sus propias tarjetas para los comandos, como `twirl()` o `jump()`.



Página de introducción



Página del playground



Comandos



Descripción general

Lección 1: Rutinas diarias

- Explorar: Debate que relaciona el horneado con los comandos
- Descubrir: Actividad Rutinas diarias
- Jugar: Crea comandos y Agrega un nuevo comando

Lección 2: Orden de la historia

- Explorar: Discusión que relaciona el orden de las tramas de la historia con los comandos
- Descubrir: Actividad Orden de la historia
- Jugar: Juego 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 Pasos básicos

Los estudiantes podrán hacer lo siguiente

- 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:** el código que le dice a una app que realice una acción específica
- **Error:** un error en el código
- **Depurar:** buscar y corregir errores en el código

Normas

1A-AP-08, 1A-AP-10, 1A-AP-12, 1A-AP-14, 1B-AP-16 >

Explorar

Objetivo: Presentar el concepto de comandos al relacionarlo con hornear brownies.

Debate:

- ¿Seguirán una receta de brownies?
- ¿Seguirán los pasos de la receta en orden?

Conclusión: Cada paso o instrucción de una receta es como un comando en programación. Pídeles a los estudiantes que elaboren sus propios comandos.

Descubrir

Objetivo: Modelar el proceso de una rutina diaria al identificar las instrucciones paso a paso.

Materiales: Tarjetas de Lavarse las manos

Instrucciones:

1. Baraja el mazo de tarjetas de Lavarse las manos y colócalas sobre una mesa o en la pizarra. Las tarjetas no deben estar en orden.
2. Pregúntales a los estudiantes si creen que hay un error en su secuencia de lavado de manos.
3. Pídeles que depuren (o corrijan) las instrucciones al mover las tarjetas a su ubicación correcta de a una a la vez.

Alternativa:

Pídeles a los estudiantes que trabajen en parejas o en grupos pequeños y entrega a cada grupo una serie de tarjetas.

Extensión:

Pídeles a los estudiantes que elaboren su propia serie de instrucciones paso a paso para algo que hacen todos los días y que dibujen los pasos específicos.

↓ [Descarga las tarjetas de Lavarse las manos](#)



Reproducir

Objetivo: Los estudiantes podrán agregar los comandos en el orden correcto para recolectar sus primeras gemas en Aprender a programar 1 en la app Swift Playgrounds.

Instrucciones:

1. Proyecta la página de introducción del capítulo “Comandos” en el playground Aprender a programar 1 en una pantalla.
2. Introducción:
 - Lean las páginas juntos como clase, deteniéndose para responder preguntas, si es necesario.
3. Crea comandos:
 - Revisa los dos comandos que los estudiantes necesitarán para llevar Byte a la gema, `moveForward()` y `collectGem()`.
 - Pídeles a los estudiantes que experimenten formas de dirigir a Byte desde la flecha de inicio hasta la gema y recolectarla. 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. Toca el botón Ejecutar mi código o haz clic en él.
 - Prueba varias ideas diferentes.
 - ¡Celebra con Byte!

Extensión:

Si los estudiantes están listos, pasa a la página siguiente, “Agrega un nuevo comando”. Aquí, los estudiantes usarán un nuevo comando, `turnLeft()`.



Aprender a programar 1

Materiales del facilitador:

- iPad o Mac
- App Swift Playgrounds
- Playground Aprender a programar 1
- Proyector o pantalla

Materiales del estudiante:

- Hojas de trabajo Crea comandos y Agrega un nuevo comando
- Lápices
- Papel adicional (opcional)



[Descarga las hojas de trabajo de Aprender a programar](#)



Explorar

Objetivo: Explorar cómo los libros siguen una secuencia (principio, medio y final) para que las historias tengan sentido.

Debate:

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

Conclusión: Establece la relación con el código y enfatiza lo importante que es colocar 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 estudiantes podrán colocar las imágenes en orden para recrear la historia con precisión.

Materiales del facilitador:

- Pizarra
- Marcadores

Materiales del estudiante:

- Papel
- Marcadores o lápices de colores
- Alternativa: dispositivo iPad y una app de dibujo

Instrucciones:

1. Lee una historia que los estudiantes conozcan bien. Como clase, determinen los puntos principales de la trama de la historia. Lo ideal sería crear de cuatro a seis puntos de la trama.
2. Crea grupos pequeños que tengan la misma cantidad de estudiantes que puntos de la trama; por ejemplo, cuatro puntos de la trama equivalen a cuatro estudiantes por grupo.
3. Pídele a cada estudiante del grupo que dibuje uno de los puntos de la trama.
4. Los grupos se turnan para estar de pie frente a la sala y los estudiantes sostienen sus dibujos de la trama desordenados.
5. El público reordena las imágenes, moviéndolas de a una a la vez.
6. Una vez que los estudiantes estén en el orden correcto, toma una foto de cada grupo.

Extensión o alternativa:

Pídele a cada grupo de estudiantes que trabaje en una historia diferente y que determinen los puntos de la trama como grupo antes de hacer los dibujos.

Reproducir

Objetivo: Los estudiantes podrán guiar a Byte a través de una cuadrícula física hasta una gema mediante comandos de orientación.

Preparación: Los estudiantes trabajarán en grupos de tres. Usa cinta de enmascarar para crear una cuadrícula de cuatro por cuatro en el suelo para cada grupo.

Instrucciones:

1. Distribuye los materiales y divide a los estudiantes en grupos de tres.
2. Lee atentamente cada rol y asigna un rol a cada persona del grupo para el primer juego.
3. Pídeles a los estudiantes que jueguen, comenzando con el rol de diseñador.
4. Jueguen tres veces y roten las tarjetas de roles cada vez.

Funciones:

- Diseñador: coloca la gema y la flecha de inicio en la cuadrícula.
- Programador: con la ayuda de tus compañeros, coloca las tarjetas de comando en la cuadrícula o junto a ella para dirigir a Byte a la gema y recolectarla.
- Evaluador: con Byte en la flecha, sigue las tarjetas de comando para moverlo por la cuadrícula. Si recolectas la gema, ¡celebra! Si no lo logras, trabajen en equipo para depurar o corregir el código.

Alternativa:

Si los estudiantes trabajan contigo de forma individual o están aprendiendo en casa, pueden jugar este juego por su cuenta en la actividad alternativa descargable de Keynote.

Materiales del facilitador:

- Cinta de enmascarar

Materiales del estudiante:

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

↓ [Descarga los materiales](#)

↓ [Descarga la actividad alternativa](#)



Explorar

Objetivo: Explorar la idea de que programar puede ser creativo.

Debate:

- Pregúntales a los estudiantes si alguna vez aprendieron un baile.
- ¿El baile tenía un orden de pasos a seguir?
- ¿Cómo sabían qué hacer a continuación?
- ¿Los pasos de baile tenían nombres?
- ¿Alguna vez usaron los mismos pasos en diferentes momentos en un baile o en diferentes rutinas de baile?

Conclusión: Ayuda a los estudiantes a apreciar que la programación es creativa y que, al igual que la coreografía de un baile, los programadores pueden crear nuevos comandos y luego juntarlos de formas diferentes e interesantes.

Descubrir

Objetivo: Crear una rutina de baile corta, junto con tarjetas para representar los pasos de baile. Cada tarjeta de Pasos de baile es como un comando en el playground de Aprender a programar.

Materiales del estudiante:

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

Instrucciones:

1. Pídeles a los estudiantes que, en parejas o en grupos pequeños, creen una rutina de baile corta.
2. Una vez que los estudiantes establezcan la rutina, crearán tarjetas con los diferentes pasos de baile. Los estudiantes deben incluir un dibujo y el nombre del paso en cada tarjeta, y deben ser lo más creativos y absurdos que sea posible.
3. Haz que cada grupo ejecute su baile y, luego, ¡hagan una fiesta de baile todos juntos!

Alternativa:

Los estudiantes pueden usar las tarjetas descargables de Pasos de baile a continuación para crear su baile o pueden usar las tarjetas como ejemplos para hacer las suyas propias.

Extensión:

Los estudiantes harán un video de su baile para mostrárselo al grupo.

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



Reproducir

Objetivo: Crear una secuencia de pasos para enseñarle a MeeBot un baile nuevo.

Instrucciones:

1. Proyecta el playground El baile de MeeBot en una pantalla. Tendrás que suscribirte a playground si todavía no lo hiciste.
2. Introducción:
 - Lean las páginas juntos como clase, deteniéndose para responder preguntas, si es necesario.
3. Hola, MeeBot:
 - Toca el botón Ejecutar mi código o haz clic en él y mira cómo baila el robot.
4. Movimientos básicos:
 - En grupo, en parejas o individualmente en su propio iPad, los estudiantes elegirán ocho comandos de la lista de sugerencias y verán bailar al robot.
 - Pídeles a los estudiantes que compartan sus bailes o creen algunos diferentes como clase.
 - ¡Baila con el robot!

Extensión:

- Pasa a la página siguiente, “Rutina de baile”, donde los estudiantes pueden agregar movimientos dentro de la función `myDanceRoutine()`. Pueden agregar tantos o tan pocos comandos como deseen.



El baile de MeeBot

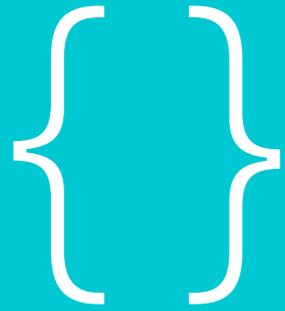
Materiales del facilitador:

- iPad o Mac
- App Swift Playgrounds
- Playground El baile de MeeBot
- Proyector o pantalla

Materiales del estudiante:

- Dispositivos iPad (opcional)





Funciones



Descripción general

Lección 1: Gema de papel

- Explorar: Debate sobre las instrucciones paso a paso
- Descubrir: Actividad Gema de papel
- Jugar: Nuevos comportamientos y Crear una nueva función

Lección 2: Festival de canciones

- Explorar: Debate sobre cómo nombrar una función
- Descubrir: Actividad Festival de canciones
- Jugar: Juego Rompecabezas en el suelo

Lección 3: Mi función relajante

- Explorar: Debate sobre resolver problemas de varias maneras
- Descubrir: Actividad Mi función relajante
- Jugar: Recolecta, activa y repite

Los estudiantes podrán hacer lo siguiente

- Desmontar un problema o tarea grande 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:** un conjunto de comandos con nombre que se puede ejecutar cuando sea necesario
- **Activar:** encender o apagar

Normas

1A-AP-08, 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-14, 1B-AP-16 >

Explorar

Objetivo: Explorar la idea de empaquetar una serie de comandos y darles un nombre.

Debate: Elijan una rutina diaria en la que enfocarse como clase. Pídeles a los estudiantes que identifique el nombre de su rutina diaria y los pasos que la conforman.

Ejemplo: Rutina para dormir

- Paso 1: Cepíllate los dientes
- Paso 2: Usa el baño
- Paso 3: Lee
- Paso 4: Di “Buenas noches”
- Paso 5: Apaga las luces

Conclusión: Crear un conjunto de instrucciones y darles un nombre es el mismo concepto que crear una función.

Extensión: Pregúntales a los estudiantes si las instrucciones de alguno de sus pasos podrían ser más específicas. Por ejemplo, ¿cuáles son los pasos específicos para cepillarse los dientes?

Descubrir

Objetivo: Para comenzar, los estudiantes seguirán las instrucciones para hacer una gema de papel y, luego, escribirán o dibujarán las instrucciones para hacer otra forma que elijan.

Materiales del estudiante:

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

Instrucciones:

Muéstrales a los estudiantes cómo hacer una gema de papel:

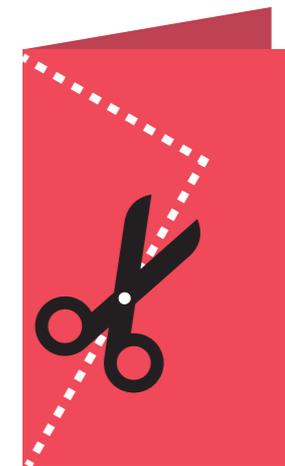
1. Dobra una hoja de papel por la mitad.
2. Dibuja una línea desde la esquina superior del lado doblado hasta una pulgada o dos por encima del centro del papel.
3. Dibuja otra línea desde donde termina la primera línea hasta la esquina inferior del lado doblado.
4. Corta a lo largo de las líneas que dibujaste.
5. Quita la gema del papel y desdóblala.

Pídeles a los estudiantes que hagan sus propias formas:

1. Divide a los estudiantes en grupos pequeños.
2. Pídeles a los grupos que decidan qué forma hacer.
3. Concédeles tiempo a los estudiantes para que practiquen cómo hacer la forma una o dos veces.
4. Pídeles que escriban o dibujen las instrucciones para hacer la forma y que luego asignen un nombre a sus instrucciones, como “Hacer un círculo” o “La letra T”.

Alternativa:

Pídeles a los estudiantes que hagan un video en el que muestren cómo hacer las formas.



Reproducir

Objetivo: Al trabajar como grupo completo, los estudiantes podrán desglosar los pasos necesarios para llevar a Byte a la gema.

Instrucciones:

1. Proyecta el playground Aprender a programar 1 en una pantalla. Dirígete al capítulo “Funciones” en Aprender a programar 1.
2. Introducción:
 - Lean las páginas juntos como clase, deteniéndose para responder preguntas, si es necesario.
3. Nuevos comportamientos:
 - Repasa los comandos `moveForward()`, `turnLeft()` y `collectGem()`, y ten en cuenta que no tienes un comando `turnRight()`.
 - Pídeles a los estudiantes que experimenten formas de dirigir a Byte desde la flecha de inicio hasta la gema y recolectarla. Los estudiantes registrarán 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. Toca el botón Ejecutar mi código o haz clic en él.
 - Prueba varias ideas diferentes.
 - ¡Celebra con Byte!
4. Crea una nueva función:
 - En base a lo que aprendieron en la última página del playground, “Nuevos comportamientos”, pídeles a los estudiantes que propongan ideas para crear la función `turnRight()`.
 - Pídeles a los estudiantes que experimenten diferentes formas de dirigir a Byte, con la función `turnRight()`, desde la flecha de inicio hasta el interruptor cerrado y activarlo.
 - Recopila ideas de la clase y escribe el código en la app Swift Playgrounds para completar el rompecabezas. Toca el botón Ejecutar mi código o haz clic en él.
 - Prueba varias ideas diferentes.
 - ¡Celebra con Byte, este rompecabezas fue difícil!



Aprender a programar 1

Materiales del facilitador:

- iPad o Mac
- App Swift Playgrounds
- Playground Aprender a programar 1
- Proyector o pantalla

Materiales del estudiante:

- Hojas de trabajo Nuevos comportamientos y Crea una nueva función
- Lápices
- Papel adicional (opcional)



[Descarga las hojas de trabajo de Aprender a programar](#)

Explorar

Objetivo: Aplicar el conocimiento de los comandos y las funciones a las canciones al darles nombres descriptivos.

Discusión: Pídeles a los estudiantes que creen una variedad de canciones y asigna a cada una un nombre de función descriptivo.

Ejemplo: En la canción “Estrellita, dónde estás” la llamada de la función podría ser `singTwinkle()`, pero `singSong1()` no sería un buen nombre porque la primera canción podría cambiar.

Conclusión: Es importante asignar nombres descriptivos a las funciones porque esto hace que el código sea más fácil de entender para ti y para otras personas.

Descubrir

Objetivo: Los estudiantes crearán un concierto al llamar a diferentes comandos de canciones en una función de concierto.

Materiales del facilitador:

- iPad o Mac
- Proyector o pantalla
- Pizarra
- Marcadores

Instrucciones:

1. Ayuda a los estudiantes a crear nombres de funciones para varias canciones, por ejemplo, `singHappyBirthday()`.
2. Como grupo, elijan el orden en el que cantar las canciones.
3. Escriban una definición de función para un concierto y completen la función con los comandos de canciones.

Ejemplo:

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

Alternativa:

Los estudiantes cantan en grupos pequeños y cada grupo elabora su propia lista de canciones, los nombres de las funciones de las canciones y el orden en el cual cantarlas. Luego, cada grupo interpreta sus canciones y hace un video de su concierto.

Reproducir

Objetivo: Los estudiantes resolverán una ecuación simple, colocarán una gema en la respuesta y luego guiarán a Byte a través de la cuadrícula con los comandos de orientación.

Preparación: Los estudiantes trabajarán en grupos de tres. Usa cinta de enmascarar 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 cuadrado restante.

Instrucciones:

1. Distribuye los materiales y divide a los estudiantes en grupos de tres.
2. Lee atentamente cada rol y asigna un rol a cada persona del grupo para el primer juego.
3. Pídeles a los estudiantes que jueguen, comenzando con el rol de diseñador.
4. Jueguen tres veces y roten las tarjetas de roles cada vez.

Funciones:

- Diseñador: tira dos dados. Con la ayuda de tus compañeros, suma los dos números y coloca la gema en el cuadrado de la cuadrícula que tenga esa suma.
- Programador: con la ayuda de tus compañeros, coloca las tarjetas de comando en la cuadrícula o junto a ella para dirigir a Byte a la gema y recolectarla.
- Evaluador: con Byte en la flecha, sigue las tarjetas de comando para moverlo por la cuadrícula. Si recolectas la gema, ¡celebra! Si no lo logras, trabajen en equipo para corregir el código.

Alternativa:

Si los estudiantes trabajan contigo de forma individual o están aprendiendo en casa, pueden jugar este juego por su cuenta en la actividad alternativa descargable de Keynote.

Materiales del facilitador:

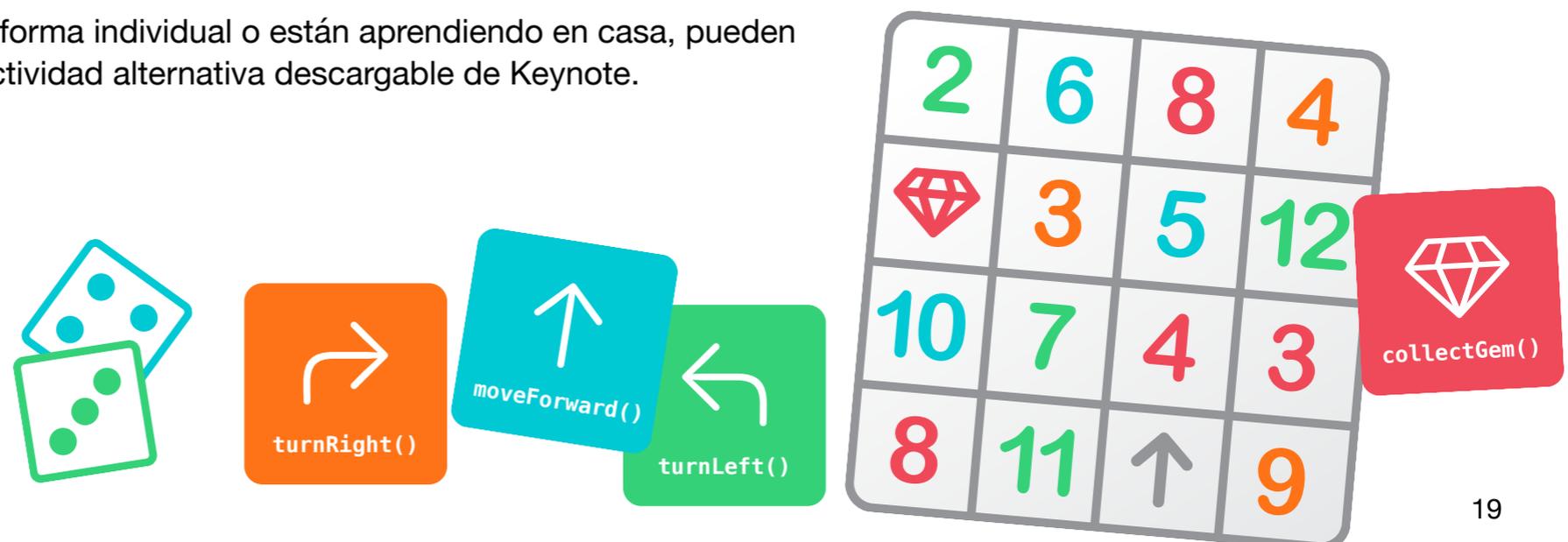
- Cinta de enmascarar
- Un conjunto de números impresos para cada cuadrícula

Materiales del estudiante:

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

↓ [Descarga los materiales](#)

↓ [Descarga la actividad alternativa](#)

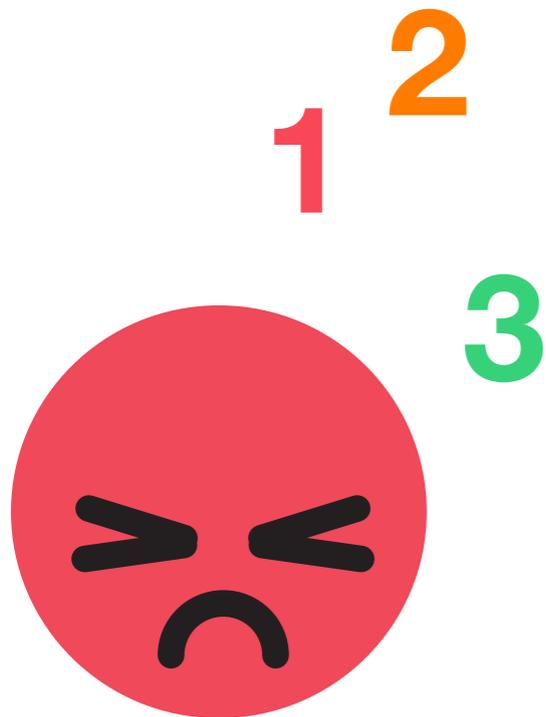


Explorar

Objetivo: Los estudiantes entenderán que, por lo general, hay más de una forma de resolver un problema.

Debate: Pídeles a los estudiantes que piensen en un problema que tuvieron y que luego compartan las maneras en que lo resolvieron. Pregunta al grupo si alguien resolvería ese problema de una manera diferente. Explora varios problemas y soluciones diferentes.

Conclusión: Ayuda a los estudiantes a hacer conexiones con el código y a aprender que, por lo general, hay más de una forma de resolver un problema de programación.



Descubrir

Objetivo: Los estudiantes escribirán una función para su técnica de relajación y le darán un nombre.

Materiales del estudiante:

- Hoja de trabajo Mi función relajante
- Lápices
- Lapiceras o lápices de colores

Instrucciones:

Consejo: Si es posible, es mejor que los estudiantes trabajen en esta actividad de forma individual.

1. Pídeles a los estudiantes que hagan una lluvia de ideas sobre formas de relajarse cuando están molestos, ya sea en casa o en la escuela. Diles que dividan en pasos sus técnicas de relajación.
2. Distribuye la hoja de trabajo Mi función relajante y pide a los estudiantes que dibujen los pasos de su técnica de relajación.
3. Los estudiantes le darán un nombre a su técnica de relajación. Pueden usar mayúsculas y minúsculas, por ejemplo, `cuentaHastaDiez()`, o simplemente usar una oración corta, como “Cuenta hasta diez”.

Extensiones:

Sin conexión: Pídeles a los estudiantes que interpreten su técnica de relajación en grupos pequeños o frente a la clase.

Con el iPad: Indícales a los estudiantes que hagan un video de su técnica de relajación para compartir con la clase.



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

Reproducir

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

Instrucciones:

1. Proyecta la página “Recolecta, activa y repite” en el playground Aprender a programar 1 en una pantalla y señala la función vacía que los estudiantes ayudarán a completar.
2. Recolecta, activa y repite:
 - Repasa los comandos `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` y `toggleSwitch()`.
 - Pídeles a los estudiantes que intenten identificar las partes del rompecabezas que se repiten y, luego, usen sus ideas para completar la función en la app y ponerle un nombre.
 - Pídeles 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 estudiantes experimentarán formas de dirigir a Byte para que recolecte todas las gemas y active todos los interruptores. Los estudiantes registrarán 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. Toca el botón Ejecutar mi código o haz clic en él.
 - Prueba varias soluciones diferentes.
 - ¡Celebren juntos, este rompecabezas fue difícil!



Aprender a programar 1

Materiales del facilitador:

- iPad o Mac
- App Swift Playgrounds
- Playground Aprender a programar 1
- Proyector o pantalla

Materiales del estudiante:

- Hoja de trabajo Recolecta, activa y repite
- Lápices
- Papel adicional (opcional)


[Descarga la hoja de trabajo de Aprender a programar](#)





Ciclos



Descripción general

Lección 1: Pétalos repetidos

- Explorar: Debate sobre los pasos repetitivos en programación y en la vida real
- Descubrir: Actividad Pétalos repetidos
- Jugar: Ciclos y Ciclos por todos lados

Lección 2: Carrera de obstáculos

- Explorar: Debate sobre los puntos de detención en un ciclo
- Descubrir: Actividad Carrera de obstáculos
- Jugar: Juego Rompecabezas en el suelo

Lección 3: Patrones de percusión

- Explorar: Debate sobre los ciclos en la música
- Descubrir: Actividad Patrones de percusión
- Jugar: Hacia afuera y hacia adentro y Ciclos de baile

Los estudiantes podrán hacer lo siguiente

- Identificar un ciclo en el código
- Desmontar un problema o tarea grande en pasos más pequeños
- Crear una secuencia de comandos y repetirla mediante un ciclo
- Probar y depurar las instrucciones y el código

Vocabulario

- **Ciclo:** un bloque de código que se repite una cierta cantidad de veces

Normas

1A-CS-01, 1A-AP-08, 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-14 >

Explorar

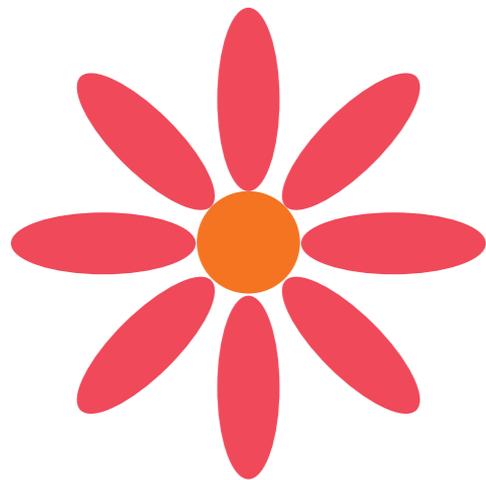
Objetivo: Relacionar la idea de los ciclos con la vida real.

Debate: Explora los momentos en los que los estudiantes podrían repetir una tarea o un paso en la vida real.

Ejemplos:

- Caminar
- Andar en bicicleta
- Coser, tejer o tejer a crochet

Conclusión: Los ciclos repiten un comando o un conjunto de comandos tantas veces como especifiques.



Descubrir

Objetivo: Los estudiantes comenzarán a explorar el concepto de ciclos al crear una flor única.

Materiales del estudiante:

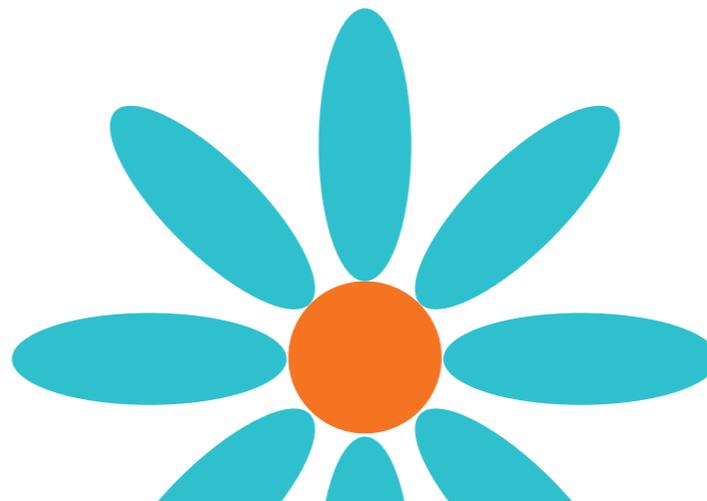
- Hoja de trabajo Pétalos repetidos
- Papel de color
- Lápices
- Tijeras
- Barras de pegamento
- Dados

Instrucciones:

1. Los estudiantes dibujan un solo pétalo, aproximadamente del largo de la palma de la mano, en una hoja de papel de color y lo recortan. Esta será su plantilla de pétalos para su flor.
2. Luego, cada estudiante lanza dos dados, suma los números y completa el número que falta en el ciclo en su hoja de trabajo Pétalos repetidos. Este es el número de pétalos que tendrá su flor.
3. Con su plantilla de pétalos, los estudiantes trazan su pétalo en papel de color y recortan el número correcto de pétalos para su flor.
4. Con la hoja de trabajo Pétalos repetidos, los estudiantes arman su flor y pegan las partes en su lugar.



[Descarga la hoja de trabajo Pétalos repetidos](#)



Reproducir

Objetivo: Los estudiantes podrán escribir código dentro de un ciclo para recolectar todas las gemas.

Instrucciones:

1. Proyecta en una pantalla el capítulo “Ciclos for” de la página de introducción del playground Aprender a programar 1.
2. Introducción:
 - Lean las páginas juntos como clase, deteniéndose para responder preguntas, si es necesario.
3. Ciclos:
 - Muéstrales a los estudiantes cómo funcionan los portales y repasa los comandos `moveForward()`, `turnLeft()`, `turnRight()` y `collectGem()`.
 - Pídeles a los estudiantes que experimenten formas de dirigir a Byte desde la flecha de inicio hasta las gemas y recolectarlas; observen qué comandos se repiten. Los estudiantes registrarán 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 indicarle a Byte que recolecte la primera gema y camine hasta el portal.
 - Pregúntales a los estudiantes cuántas gemas hay y agrega ese número al ciclo. Toca el botón Ejecutar mi código o haz clic en él.
 - Prueba varias soluciones diferentes.
 - ¡Celebra con Byte!
4. Ciclos por todos lados:
 - Pídeles a los estudiantes que experimenten formas de recolectar todas las gemas y que observen qué comandos se repiten.
 - Para agregar un ciclo `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 Swift Playgrounds para completar el rompecabezas. Toca el botón Ejecutar mi código o haz clic en él.
 - Prueba varias ideas diferentes.
 - ¡Celebra con Byte!



Aprender a programar 1

Materiales del facilitador:

- iPad o Mac
- App Swift Playgrounds
- Playground Aprender a programar 1
- Proyector o pantalla

Materiales del estudiante:

- Hojas de trabajo Ciclos y Ciclos por todos lados
- Lápices
- Papel adicional (opcional)



[Descarga las hojas de trabajo de Aprender a programar](#)

Explorar

Objetivo: Explorar por qué los ciclos siempre necesitan un punto final específico.

Debate: Pídeles a los estudiantes que imaginen una rueda de la fortuna u otra atracción con la que estén familiarizados. ¿Qué pasaría si el operador no presionara el botón para detenerla después de cinco rondas? Pídeles que piensen otros ejemplos de lo que sucedería si no se detiene un ciclo.

Conclusión: Ayuda a los estudiantes a entender que, si no detienen un ciclo, se repetirá infinitamente.

Descubrir

Objetivo: Los estudiantes descubrirán cómo funcionan los ciclos al ejecutarlos en una carrera de obstáculos que ellos mismos hayan diseñado.

Materiales:

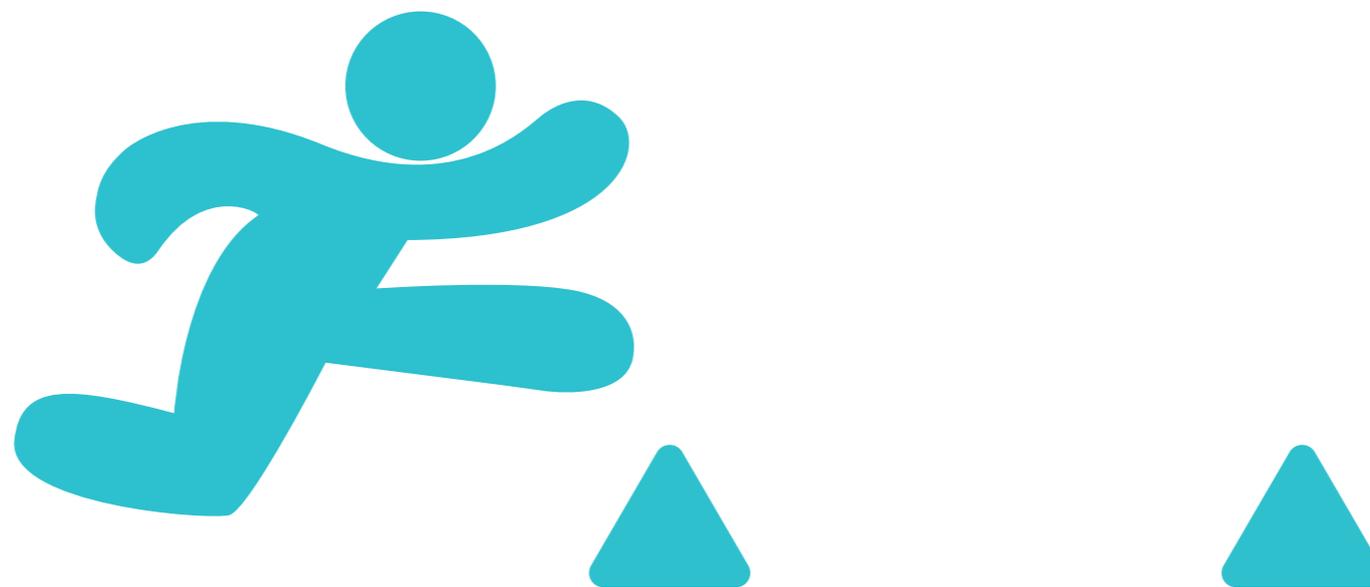
- Espacio para hacer la actividad física
- Accesorios de la carrera de obstáculos
- Dado

Instrucciones:

1. Crea una carrera de obstáculos corta en tu aula o al aire libre.
2. Lanza un dado y pídeles a los estudiantes que repitan la carrera tantas veces como indique el dado.

Alternativa:

Los estudiantes proponen una serie de movimientos, por ejemplo, tocarse los dedos de los pies, saltar, patear con una pierna. Luego de lanzar un dado, los estudiantes repiten la serie de movimientos el número de veces que indica el dado.



Reproducir

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

Preparación: Los estudiantes trabajarán en grupos de tres. Usa cinta de enmascarar para crear una cuadrícula de cuatro por cuatro en el suelo para cada grupo.

Instrucciones:

1. Distribuye los materiales y divide a los estudiantes en grupos de tres.
2. Lee atentamente cada rol y asigna un rol a cada persona del grupo para el primer juego.
3. Pídeles a los estudiantes que jueguen, comenzando con el rol de diseñador.
4. Jueguen tres veces y roten las tarjetas de roles cada vez.

Funciones:

- Diseñador: con la ayuda de tus compañeros, coloca tres gemas en un patrón repetitivo en la cuadrícula. Coloca la flecha de inicio en la cuadrícula.
- Programador: con la ayuda de tus compañeros, coloca las tarjetas de comando en la cuadrícula o junto a ella para dirigir a Byte hasta las gemas y recolectarlas. Usa las tarjetas de Ciclo para decirle al evaluador cuántas veces ejecutar el ciclo a través de los comandos.
- Evaluador: con Byte en la flecha, sigue las tarjetas de comando para moverlo por la cuadrícula. Si recolectas todas las gemas, ¡celebra! Si no lo logras, trabajen en equipo para corregir el código.

Alternativa:

Si los estudiantes trabajan contigo de forma individual o están aprendiendo en casa, pueden jugar este juego por su cuenta en la actividad alternativa descargable de Keynote.

Materiales del facilitador:

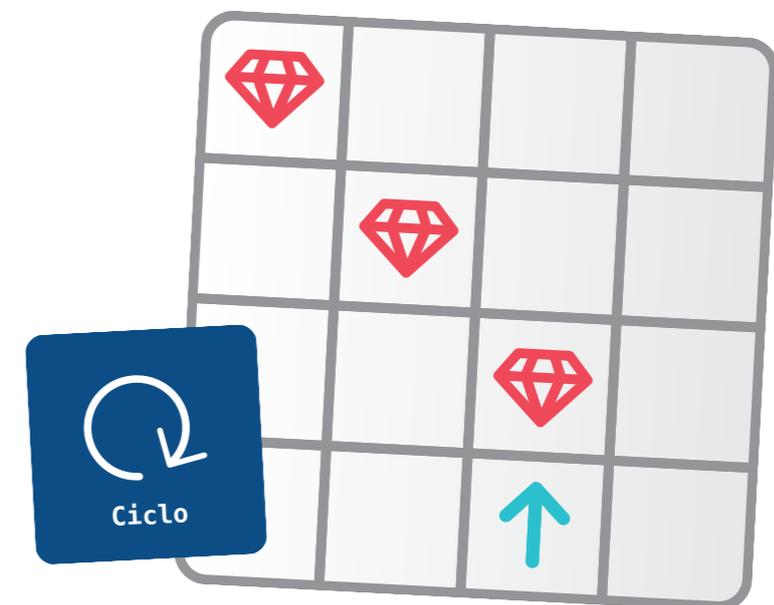
- Cinta de enmascarar

Materiales del estudiante:

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

↓ [Descarga los materiales](#)

↓ [Descarga la actividad alternativa](#)



Explorar

Objetivo: Explorar los patrones repetitivos en la música.

Debate: Pídeles a los estudiantes que conversen sobre los instrumentos que tocan o las canciones que cantan. Pregúntales si alguna vez repiten un ritmo o un estribillo cuando tocan o cantan. ¿Pueden pensar en otras partes de una canción o de la música que se repiten?

Conclusión: Refuerza la idea de que los ciclos consisten en dos partes:

- Los comandos
- La cantidad de veces que se repiten



Descubrir

Objetivo: Los estudiantes podrán repetir un patrón de batería al hacer una conexión entre el código del ciclo y un ejemplo físico de la vida real.

Materiales:

- Algo sobre lo que se pueda hacer percusión, como el suelo, los muslos o libros
- Espacio para sentarse en círculo

Instrucciones:

1. Pídeles a los estudiantes que se sienten en círculo.
2. Diles que repitan el ritmo de redoble que creaste las veces que indiques levantando los dedos. Por ejemplo, si levantas cuatro dedos, los estudiantes deben repetir el redoble cuatro veces y luego detenerse.
3. Túrnense para dar la vuelta al círculo o divídanse en grupos pequeños para que cada estudiante tenga la oportunidad de ser el baterista principal.

Extensión:

Pídeles a los estudiantes que hagan una batería.

Reproducir

Objetivo: Los estudiantes llamarán a varios comandos diferentes dentro de un ciclo y determinarán cuántas veces se debe llamar al ciclo.

Instrucciones:

1. Proyecta el playground Aprender a programar 1 en una pantalla. Dirígete a la tercera página, “Hacia afuera y hacia adentro”, del capítulo “Ciclos ‘for’”.
2. Hacia afuera y hacia adentro:
 - Repasa los comandos `moveForward()`, `turnLeft()`, `turnRight()`, `collectGem()` y `toggleSwitch()`.
 - Pídeles a los estudiantes que experimenten formas de dirigir a Byte desde la flecha de inicio hasta cada interruptor cerrado y activarlo.
 - Para agregar un ciclo `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. Toca el botón Ejecutar mi código o haz clic en él.
 - Prueba varias ideas diferentes.
 - ¡Celebra con Byte!
3. Sal de Aprender a programar 1 y abre el playground El baile de MeeBot, y dirígete a la página “Ciclos de baile”. (No hay una hoja de trabajo para el estudiante en esta página del playground).
4. Ciclos de baile:
 - Pídeles a los estudiantes que trabajen en grupo, en parejas o individualmente en su propio iPad para completar el ciclo y ver bailar al robot.
 - Pídeles a los estudiantes que compartan sus bailes o creen algunos diferentes como clase.
 - ¡Baila con el robot!



Aprender a programar 1



El baile de MeeBot

Materiales del facilitador:

- iPad o Mac
- App Swift Playgrounds
- Playground Aprender a programar 1
- Playground El baile de MeeBot
- Proyector o pantalla

Materiales del estudiante:

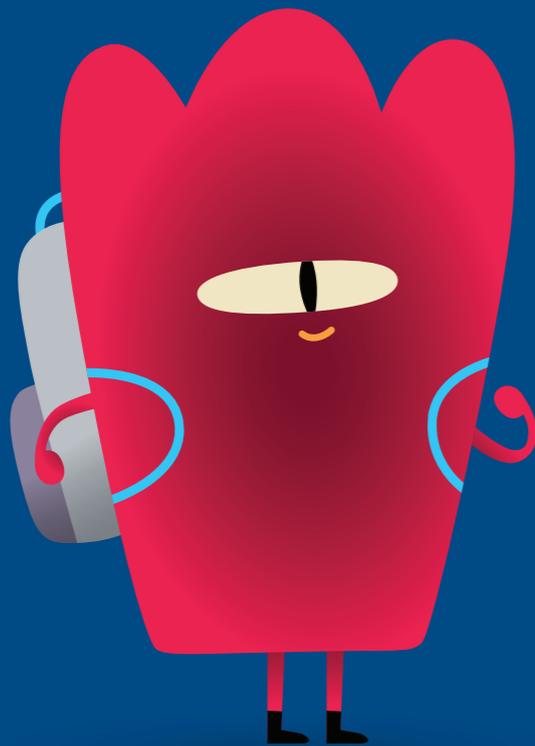
- Hoja de trabajo Hacia afuera y hacia adentro
- Lápices
- Dispositivos iPad (opcional)
- Papel adicional (opcional)



[Descarga la hoja de trabajo de Aprender a programar](#)



Variables



Descripción general

Lección 1: ¿Se hunde o flota?

- Explorar: Debate sobre actualizar una variable
- Descubrir: Actividad ¿Se hunde o flota?
- Jugar: Hacer un seguimiento y Juego de muestra

Lección 2: Juego de palabras

- Explorar: Debate sobre tipos de respuestas a preguntas
- Descubrir: Actividad Juego de palabras
- Jugar: Juego Rompecabezas en el suelo

Lección 3: Todo sobre mí

- Explorar: Debate sobre responder preguntas con listas
- Descubrir: Actividad Todo sobre mí
- Jugar: Usando un ciclo

Los estudiantes podrán hacer lo siguiente

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

Vocabulario

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

Normas

1A-AP-09, 1B-AP-09, 1B-AP-10, 1B-AP-16 >

Explorar

Objetivo: Explorar el concepto de variables al contar objetos y actualizar el número de la variable.

Materiales del facilitador:

- Pizarra
- Marcador
- Borrador
- Contenedor
- Cinco lápices (o cinco objetos iguales)

Instrucciones:

1. Para comenzar, escribe una instrucción de variable en la pizarra para llevar la cuenta de tus objetos.
 - Ejemplo: `var numberOfPencils = 0`
2. Sostén un contenedor vacío y diles a los estudiantes que el contenedor representa tu variable, `numberOfPencils`.
3. Agrega un lápiz al contenedor y pregunta a los estudiantes cuál es el conteo de la variable ahora. Cuando respondan correctamente, borra el `0` y escribe `1`.
4. Continúa hasta que agregues todos los lápices y tu código diga: `var numberOfPencils = 5`.
5. Luego, comienza a sacar lápices del contenedor y actualiza la variable a medida que los eliminas.

Conclusión: Ayuda a los estudiantes a entender que las variables almacenan un trozo de información. En este caso, la información es un número y el número te dice cuántos lápices hay en el contenedor.

Descubrir

Objetivo: Con los objetos encontrados, los estudiantes realizarán experimentos para determinar si se hunden o flotan; luego, registrarán los datos mediante imágenes (literales de imagen) y valores verdaderos o falsos (booleanos).

Materiales del estudiante:

- dispositivos iPad
- App Keynote
- Hoja de trabajo ¿Se hunde o flota?
- Cubeta de agua
- Varios objetos para probar

Instrucciones:

1. Divide a los estudiantes en grupos pequeños.
2. Pídeles que recolecten varios elementos para probar.
3. Diles que hagan lo siguiente con cada objeto:
 - Tomar una foto del elemento y agregarla a la hoja de trabajo.
 - Probar el elemento en el agua.
 - Registrar los resultados en la hoja de trabajo al dibujar un círculo en verdadero o falso.



[Descarga la hoja de trabajo ¿Se hunde o flota?](#)

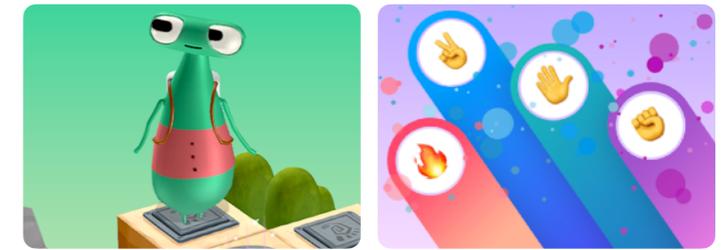
Reproducir

Objetivo: Los estudiantes podrán crear y actualizar variables en dos contextos de programación diferentes.

Instrucciones:

1. Proyecta el playground Aprender a programar 2 en una pantalla. Dirígete al capítulo “Variables”.
2. Introducción:
 - Lean las páginas juntos como clase, deteniéndose para responder preguntas, si es necesario.
3. Hacer un seguimiento:
 - Pídeles a los estudiantes que experimenten formas de dirigir a Hopper desde la flecha de inicio hasta la gema y recolectarla. Los estudiantes registrarán 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. Toca el botón Ejecutar mi código o haz clic en él.
 - Prueba varias ideas diferentes.
 - ¡Celebra con Hopper!
4. Sal de Aprender a programar 2 y pasa a la última página del playground Piedra, papel o tijeras, llamada “Juego de muestra”. (No hay una hoja de trabajo para el estudiante en esta página del playground).
5. Juego de muestra:
 - Toca el botón Ejecutar mi código o haz clic en él para jugar el juego antes de cambiar algo.
 - Decidan como grupo qué partes del juego quieren personalizar. Algunos de los cambios divertidos que pueden hacer incluyen `game.roundsToWin`, `game.challenger.emoji`, `game.addOpponent` y `game.roundPrize`.
 - Jueguen varias veces y cambien algo diferente cada vez.

Extensión: Muchas variables se establecen en el archivo `Game.swift`. Si los estudiantes sienten 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.



Aprender a programar 2

Piedra, papel o tijeras

Materiales del facilitador:

- iPad o Mac
- App Swift Playgrounds
- Playground Aprender a programar 2
- Playground Piedra, papel o tijeras
- Proyector o pantalla

Materiales del estudiante:

- Hoja de trabajo Hacer un seguimiento
- Lápices
- Papel adicional (opcional)



[Descarga la hoja de trabajo de Aprender a programar](#)

Explorar

Objetivo: Explorar varios tipos de respuestas en el mundo real y relacionarlos con diferentes tipos de Swift, que incluyen sí/no o verdadero/falso (booleanos), números (enteros), palabras (cadenas), colores (literales de color) e imágenes (literales de imagen).

Materiales del facilitador:

- Pizarra
- Marcadores

Debate: Como clase, piensen algunas preguntas que requieran diferentes tipos de respuestas y escribanlas en la pizarra.

Ejemplos:

- ¿De qué color son tus ojos? → color
- ¿Tienes una mascota? → sí/no
- ¿Tienes hermanos? → sí/no
- ¿Qué edad tienes? → número
- ¿Cómo te llamas? → word

Conclusión: Explica que también existen diferentes tipos de variables, que incluyen números, palabras, colores, imágenes y respuestas de sí o no. Según 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 estudiantes completarán un juego de palabras al rellenar el tipo de respuesta correcto.

Materiales del estudiante:

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

Instrucciones:

Pídeles a los estudiantes que trabajen en uno o más juegos de palabras en grupos pequeños. Lo ideal sería que cada grupo tenga al menos un lector o una persona de apoyo. Si ninguno de los estudiantes sabe leer, jueguen algunos juegos juntos como grupo.

Extensión: Si los estudiantes pueden hacerlo, pídeles que creen un juego de palabras para que lo complete un compañero. Anímalos a usar números, palabras, colores, imágenes y respuestas de sí o no para rellenar los espacios en blanco.



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

Reproducir

Objetivo: Los estudiantes podrán guiar a Byte para que recolecte varias gemas, agregar cada gema a un contenedor y actualizar una variable.

Preparación: Los estudiantes trabajarán en grupos de tres. Usa cinta de enmascarar para crear una cuadrícula de cuatro por cuatro en el suelo para cada grupo.

Instrucciones:

1. Distribuye los materiales y divide a los estudiantes en grupos de tres.
2. Lee atentamente cada rol y asigna un rol a cada persona del grupo para el primer juego.
3. Pídeles a los estudiantes que jueguen, comenzando con el rol de diseñador.
4. Jueguen tres veces y roten las tarjetas de roles cada vez.

Funciones:

- Diseñador: coloca varias gemas y la flecha de inicio en la cuadrícula.
- Programador: con la ayuda de tus compañeros, coloca las tarjetas de comando en la cuadrícula o junto a ella para dirigir a Byte a las gemas y recolectarlas.
- Evaluador: con Byte en la flecha, sigue los comandos para moverlo por la cuadrícula y agrega las gemas al contenedor a medida que las recolectas. Si recolectas todas las gemas, actualiza la variable `numberOfGems` en el contenedor ¡y celebra! Si no las recolectas todas, trabaja en equipo para corregir el código.

Alternativa:

Si los estudiantes trabajan contigo de forma individual o están aprendiendo en casa, pueden jugar este juego por su cuenta en la actividad alternativa descargable de Keynote.

Materiales del facilitador:

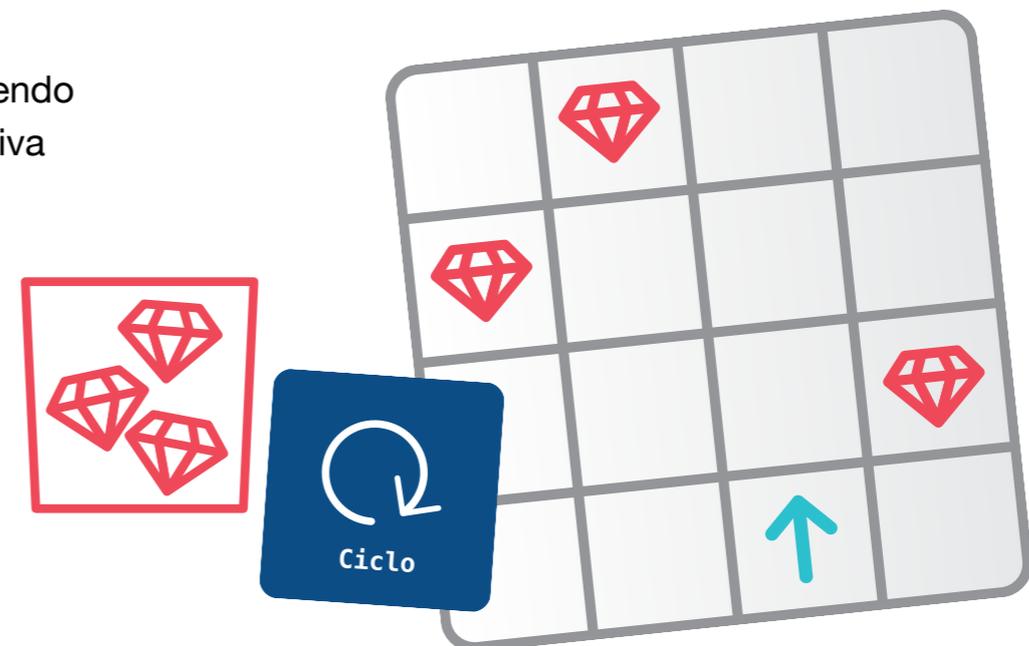
- Cinta de enmascarar

Materiales del estudiante:

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

↓ [Descarga los materiales](#)

↓ [Descarga la actividad alternativa](#)



Explorar

Objetivo: Explorar cómo usar listas, o *arreglos*, al crear variables.

Debate: ¿Qué sucedería si una hoja de trabajo les preguntara a los estudiantes el nombre de su hermano y tuvieran más de uno? Recopila ideas de la clase. Si sugieren hacer una lista, ¡diles que eso es lo que hacen los programadores! Cuando una variable tiene más de una respuesta, los estudiantes deben crear una lista.

Pídeles a los estudiantes que piensen preguntas que podrían tener varias respuestas.

Ejemplos:

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

Conclusión: Las listas que los estudiantes crean en programación son como listas en una oración.

Descubrir

Objetivo: Los estudiantes podrán completar variables para describir cosas sobre ellos mismos y sobre un compañero. Es posible que los estudiantes tengan la oportunidad de usar un arreglo como tipo de variable.

Materiales del estudiante:

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

Instrucciones:

1. Pídeles a los estudiantes que completen la hoja de trabajo Todo sobre mí.
 - Si tienen más de un hermano o mascota, pídeles que hagan una lista de elementos separados por comas.
2. Pon a los estudiantes en parejas para que completen la hoja de trabajo Todo sobre ti.

Alternativa: Los estudiantes pueden usar su iPad y la app Keynote para completar las hojas de trabajo, tomar fotos para las respuestas con imágenes y colorear los literales de color mediante las opciones de formato.



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

Reproducir

Objetivo: Los estudiantes podrán identificar una variable en programación y explorar formas en las que pueden usar los arreglos con los ciclos.

Instrucciones:

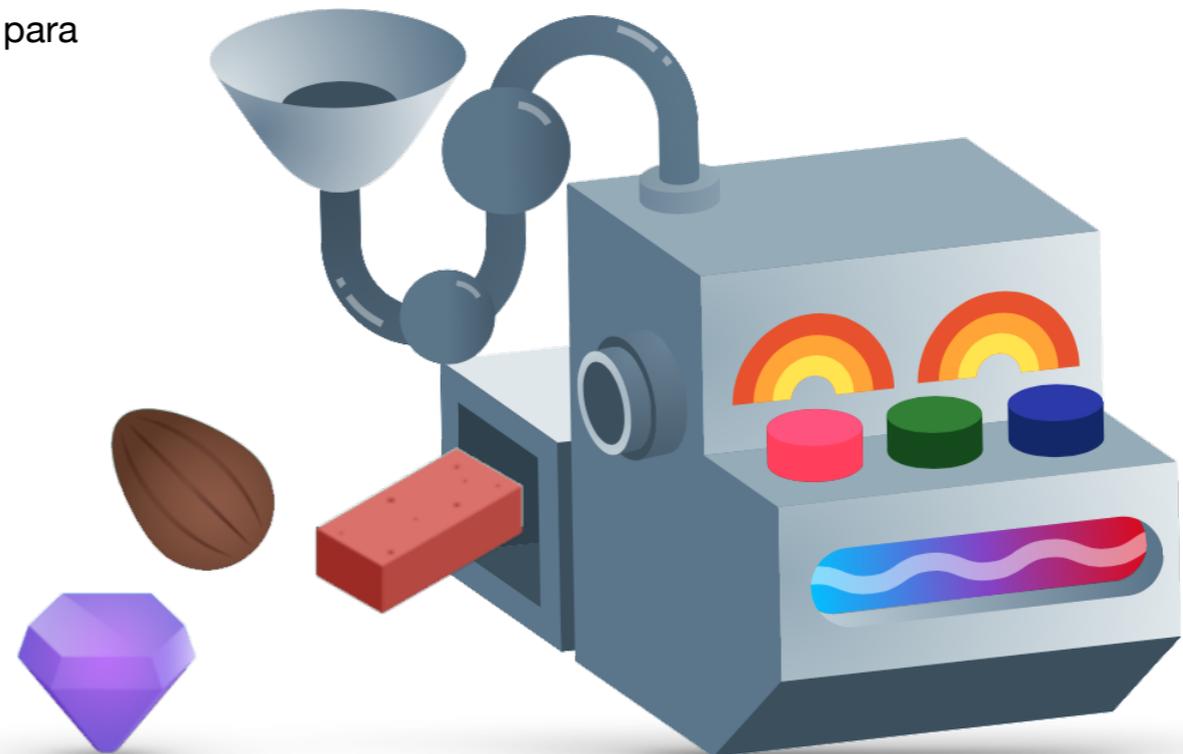
1. Proyecta el playground Máquina de código en una pantalla.
2. Introducción:
 - Lean las páginas juntos como clase, deteniéndose para responder preguntas, si es necesario.
 - Opcional: Jueguen con las primeras dos páginas: “Explorar la máquina” y “Combinar con colores”.
3. Usando un ciclo:
 - En esta página, los estudiantes combinarán sus conocimientos sobre ciclos con variables.
 - Comprueba si los estudiantes pueden identificar la variable en el código que usa un arreglo.
 - Toca el botón Ejecutar mi código o haz clic en él para ver lo que la máquina crea.
 - Continúa con el segundo paso de las instrucciones y actualiza el código para que incluya una segunda variable, elementos y un ciclo anidado. Toca el botón Ejecutar mi código o haz clic en él nuevamente para ver lo que la máquina crea.
 - Nota: Prueba esta página antes de realizar la actividad con los estudiantes.



Máquina de código

Materiales del facilitador:

- iPad o Mac
- App Swift Playgrounds
- Playground Máquina de código
- Proyector o pantalla



Diseño de una app



Explorar

Objetivo: Explorar apps familiares en diferentes dispositivos.

Instrucciones: Inicia un debate sobre las apps que los estudiantes usan en el iPad en casa o en la escuela. Luego, habla sobre las apps que ellos mismos o sus padres o tutores usan en los dispositivos de su casa.

Conclusión: Refuerza la idea de que las apps no solo se encuentran en los teléfonos, sino también en los relojes, las tabletas, las computadoras e incluso en la televisión.

Extensión: Profundiza sobre algunos ejemplos de apps y pregúntales a los estudiantes para quién está diseñada cada app, para qué sirve y por qué creen que se creó.

Ejemplo:

- App: Swift Playgrounds
- ¿Para quién es?: Personas que quieren aprender 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 poco o ningún conocimiento de programación

Descubrir

Objetivo: Preparar a los estudiantes para que diseñen sus propias apps mediante el análisis de una app familiar.

Materiales del estudiante:

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

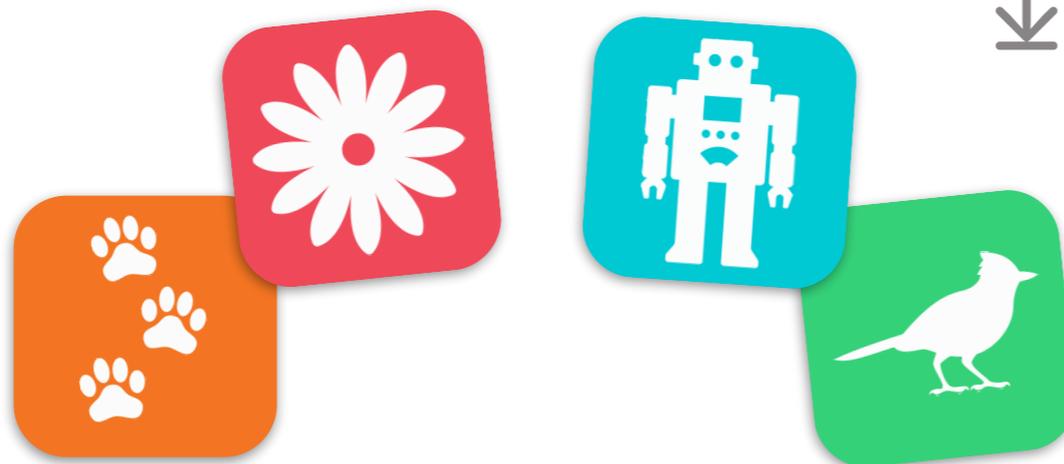
Instrucciones:

1. Divide a los estudiantes en grupos pequeños o pídeles que trabajen individualmente.
2. Pídeles que elijan una app del iPad.
3. Indícales que usen la hoja de trabajo ¿Qué es una app? para guiar su exploración de la app.
4. Invita a los estudiantes a compartir sus hallazgos sobre la app, ya sea con todo el grupo o en parejas.

Consejo para el facilitador: Cuanto más jóvenes sean tus estudiantes, más ayuda necesitarán para completar esta hoja de trabajo. En un grupo de jardín de infantes a primer grado, considera hacer dos o tres apps juntos como grupo.



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



Reproducir

Objetivo: ¡Los estudiantes diseñarán sus propias apps!

Materiales del estudiante:

- Hoja de trabajo Diseño de mi app
- Plantillas del dispositivo
- Papel adicional
- Lápices
- Lapiceras o lápices de colores

Instrucciones:

1. Divide a los estudiantes en grupos pequeños o pídeles que trabajen individualmente.
2. Guía a los estudiantes en la hoja de trabajo Diseño de mi app para orientar su proceso inicial de diseño de una app.
3. Pídeles a los estudiantes que hagan un prototipo de las páginas de sus apps con el papel adicional o las plantillas del dispositivo.
4. Indícales que creen una versión final de los prototipos de sus apps con las plantillas del dispositivo.
5. Invita a cada estudiante o grupo de estudiantes a presentar sus ideas a todo el grupo.

↓ [Descarga la hoja de trabajo Diseño de mi app](#)

↓ [Descarga las plantillas del dispositivo](#)



Recursos para el facilitador



Glosario

- **Booleano:** un tipo que tiene un valor verdadero o falso
- **Error:** un error en el código
- **Comando:** el código que le dice a una app que realice una acción específica
- **Datos:** información
- **Depurar:** buscar y corregir errores en el código
- **Función:** un conjunto de comandos con nombre que se puede ejecutar cuando sea necesario
- **Ciclo:** un bloque de código que se repite una cierta cantidad 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:** un contenedor con nombre que almacena un valor que se puede modificar

Estándares CSTA >

1A-AP

- 1A-AP-08: Modelar los procesos diarios al crear y seguir algoritmos (conjuntos de instrucciones paso a paso) para completar tareas.
- 1A-AP-09: Modelar la forma en que los programas almacenan y manipulan datos usando números u otros símbolos para representar información.
- 1A-AP-10: Desarrollar programas con secuencias y ciclos simples para expresar ideas o abordar un problema.
- 1A-AP-11: Descomponer (desglosar) los pasos necesarios para resolver un problema en una secuencia precisa de instrucciones.
- 1A-AP-12: Desarrollar planes que describan la secuencia de eventos, las metas y los resultados esperados de un programa.
- 1A-AP-14: Depurar (identificar y corregir) errores en un algoritmo o un programa que incluye secuencias y ciclos simples.

1A-CS

- 1A-CS-01: Seleccionar y operar el software apropiado para realizar una variedad de tareas y reconocer que los usuarios tienen diferentes necesidades y preferencias en cuanto a la tecnología que usan.

1B-AP

- 1B-AP-09: Crear programas que usen variables para almacenar y modificar datos.
- 1B-AP-10: Crear programas que incluyan secuencias, eventos, ciclos y condicionales.
- 1B-AP-16: Asumir diferentes roles, con la guía de un facilitador, al colaborar con compañeros durante las etapas de diseño, implementación y revisión del desarrollo del programa.

Respuestas de ejemplo

En las próximas páginas, se ofrece una posible solución para cada rompecabezas de Swift Playgrounds, pero los rompecabezas se pueden resolver de más de una manera. Anima a los estudiantes a probar diferentes maneras de dirigir a Byte u otros personajes.

Celebra todos los tipos de programación y objetivos que puedan tener los estudiantes. Es posible que algunos estudiantes quieran explorar todo el espacio del rompecabezas además de recolectar las gemas, mientras que otros quieran girar tantas veces como puedan en el camino hacia la recolección de gemas. No lo olvides, ¡la programación debería ser divertida!



Aprender a programar 1

Capítulo sobre comandos

Crea comandos

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

Capítulo sobre comandos

Agrega un nuevo comando

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

Capítulo sobre funciones

Nuevos comportamientos

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

Capítulo sobre funciones

Crea una nueva función

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

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



Aprender a programar 1

Capítulo sobre funciones

Recolecta, activa y repite

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

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

Capítulo sobre ciclos

Ciclos

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

Capítulo sobre ciclos

Ciclos por todos lados

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

Capítulo sobre ciclos

Hacia afuera
y hacia adentro

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



Aprender a programar 2

Capítulo sobre variables

Hacer un seguimiento

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



Piedra, papel o tijeras

Juego de muestra

No hay un ejemplo de solución para esta página porque el juego es completamente personalizable: ¡puedes jugarlo como quieras!



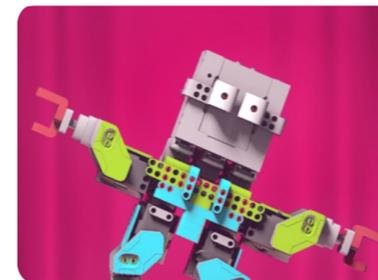
Máquina de código

Usando un ciclo

```
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()
}
```



El baile de MeeBot

Basic Moves (Pasos básicos)

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

Ciclos de baile

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

Llegar más lejos



Guías para profesores de Rompecabezas y Aventuras de Programación para todos

Las guías para profesores están diseñadas para ayudar a los educadores a enseñar programación con confianza y a profundizar el aprendizaje de los estudiantes a través de la participación en el mundo real, la comunicación, el trabajo en equipo, el pensamiento crítico y el aprendizaje personalizado. Las guías también ofrecen ideas de evaluación y consejos para diferenciar las actividades de la clase. [Descarga los libros Rompecabezas y Aventuras >](#)

Apple Teacher

Apple Teacher es un programa de aprendizaje profesional gratuito y a tu propio ritmo que ofrece acceso ilimitado a materiales de aprendizaje y contenido para usar la tecnología de Apple en educación. [Obtén más información >](#)

Apple Professional Learning

Los Apple Professional Learning Specialists lideran experiencias prácticas y envolventes que ayudan a los educadores a desarrollar prácticas didácticas innovadoras e involucrar a los estudiantes en un aprendizaje más profundo. Envía un correo electrónico a AppleProfessionalLearning@apple.com para obtener más información.



Club de programación con Swift de Programación para todos

Los clubes de programación con Swift son una excelente forma de presentar la programación en cursos para después de la escuela, campamentos de verano y otros entornos de aprendizaje informal. El diseño modular del Club de programación con Swift es ideal tanto para los programadores principiantes como para aquellos con más experiencia. [Descarga el kit de los clubes de programación con Swift >](#)



Programación: Un inicio rápido

Esta guía incluye 10 actividades de programación divertidas para niños de 10 años en adelante. Los estudiantes pueden aprender a programar en la escuela o desde casa a través de la app Swift Playgrounds, que es gratuita tanto en iPad como en Mac. [Descarga Programación: Un inicio rápido >](#)



Diario de diseño de apps

Con el Diario de diseño de apps, los estudiantes pueden poner en práctica el proceso de diseño de apps para resolver problemas en su institución educativa o comunidad. En el diario, se incentiva a los estudiantes a aportar ideas, planificar, crear prototipos y evaluar sus propias ideas de apps. Para finalizar, realizan una presentación de un prototipo de su app. [Descarga el Diario de diseño de apps >](#)

