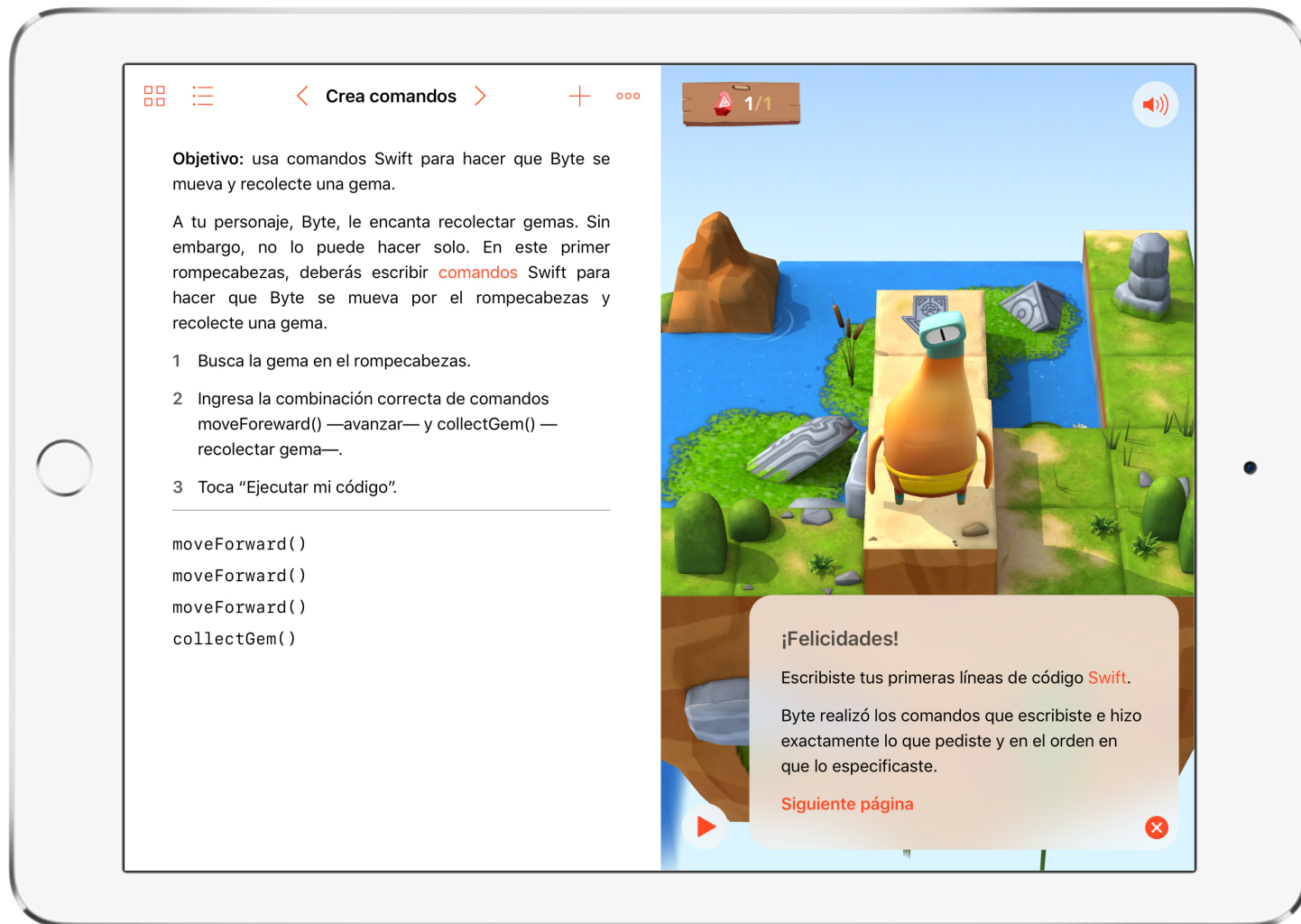




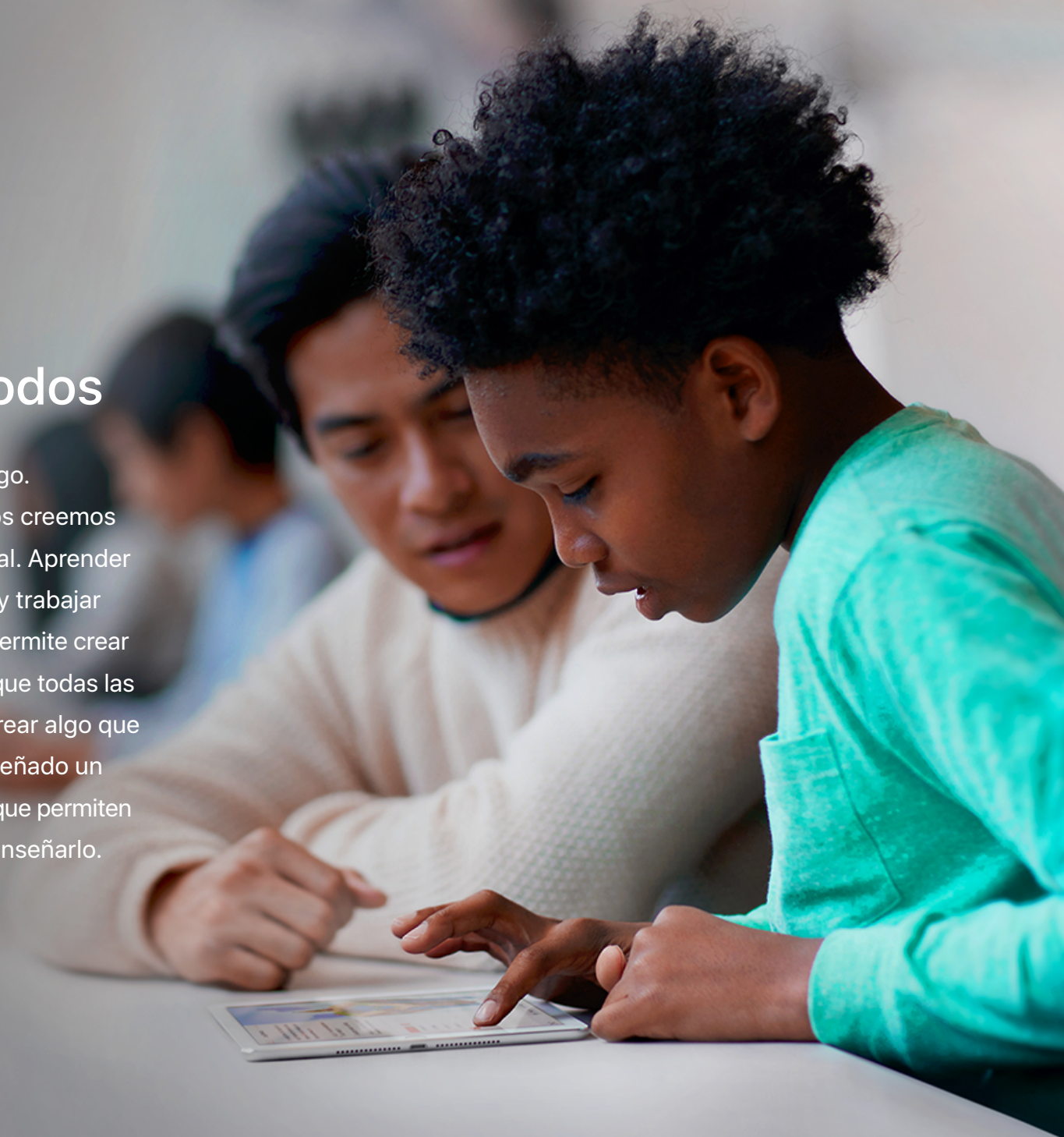
# Guía del plan de estudios de Swift Playgrounds

Febrero de 2018









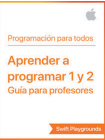


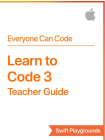








# Programación para todos

La tecnología tiene un lenguaje llamado código. El código se utiliza para programar, y nosotros creemos que la programación es una habilidad esencial. Aprender a programar te enseña a resolver problemas y trabajar en conjunto de manera creativa. Además, te permite crear apps que hacen realidad tus ideas. Creemos que todas las personas deberían tener la oportunidad de crear algo que pueda cambiar el mundo. Por eso, hemos diseñado un nuevo programa con recursos y herramientas que permiten a cualquier persona aprenderlo, escribirlo y enseñarlo.



# Plan de estudios de Programación para todos

Programación para todos incluye una variedad de recursos para que los estudiantes sin experiencia en programación aprendan a crear sus primeras apps. En la tabla que aparece a continuación, se proporciona una descripción general de todos los recursos gratuitos de enseñanza y aprendizaje disponibles.

Plan de estudios	Dispositivo	Audiencia	App	Requisitos previos	Descripción general	Materiales de aprendizaje	Recursos de apoyo	Número de horas de lección incluidas
		Desde la educación inicial hasta el segundo grado		Ninguno	Comienza a pensar como un programador explorando, en la práctica, conceptos de programación mediante apps de base visual.	<ul style="list-style-type: none"> <li>Lecciones de la app codeSpark Academy</li> <li>Curso de Cadete Espacial de Tynker</li> </ul>	Empezar a programar 1: Guía para profesores	30 horas, en las que se incluyen las lecciones de las apps y la Guía para profesores
		Desde el tercer grado hasta el quinto grado		Ninguno	Explora los conceptos fundamentales de programación y practica cómo es pensar como un programador mediante apps de base visual.	Curso Hechizos de dragón de Tynker	Empezar a programar 2: Guía para profesores	36 horas, en las que se incluyen las lecciones de las apps y la Guía para profesores
		Escuela secundaria en adelante		Ninguno	Aprende conceptos fundamentales de programación utilizando código Swift real.	<ul style="list-style-type: none"> <li>App Swift Playgrounds</li> <li>Lecciones de Aprender a programar 1 y 2</li> <li>Curso de iTunes U</li> </ul>	<ul style="list-style-type: none"> <li>Aprender a programar 1 y 2: Guía para profesores</li> <li>Distintivos de Swift Playgrounds de Apple Teacher Learning Center</li> </ul>	Hasta 85 horas, en las que se incluyen las lecciones de Aprender a programar 1 y 2, y la Guía para profesores
		Escuela secundaria en adelante		Aprender a programar 1 y 2	Amplía tus habilidades de programación y comienza a pensar como un desarrollador profesional de apps.	<ul style="list-style-type: none"> <li>App Swift Playgrounds</li> <li>Lecciones de Aprender a programar 3</li> </ul>	Aprender a programar 3: Guía para profesores	Hasta 45 horas, en las que se incluyen las lecciones de Aprender a programar 3 y la Guía para profesores
		Escuela media y superior en adelante		Ninguno	Adquiere experiencia práctica con las herramientas, las técnicas y los conceptos que se necesitan para crear una app de iOS básica desde cero.	Libro y archivos de proyectos de Introducción al desarrollo de apps con Swift	<ul style="list-style-type: none"> <li>Introducción al desarrollo de apps con Swift: Guía para profesores</li> <li>Talleres de capacitación profesional de MobileMakersEdu</li> </ul>	90 horas
		Escuela media y superior en adelante		Ninguno	Construye una base sólida en el uso de Swift, de UIKit y de redes mediante actividades prácticas y proyectos guiados. Los estudiantes podrán crear una app con diseño propio al final de este curso.	Libro y archivos de proyectos de Desarrollo de apps con Swift	<ul style="list-style-type: none"> <li>Desarrollo de apps con Swift: Guía para profesores</li> <li>Talleres de capacitación profesional de MobileMakersEdu</li> </ul>	180 horas



# Descripción general

Swift Playgrounds es una app gratuita para iPad de Apple que te permite aprender y experimentar con códigos de una manera divertida e interactiva. Los estudiantes pueden resolver juegos de ingenio para dominar los conceptos básicos de Swift, un potente lenguaje de programación creado por Apple que usan los profesionales para crear muchas de las apps más populares de hoy en día.

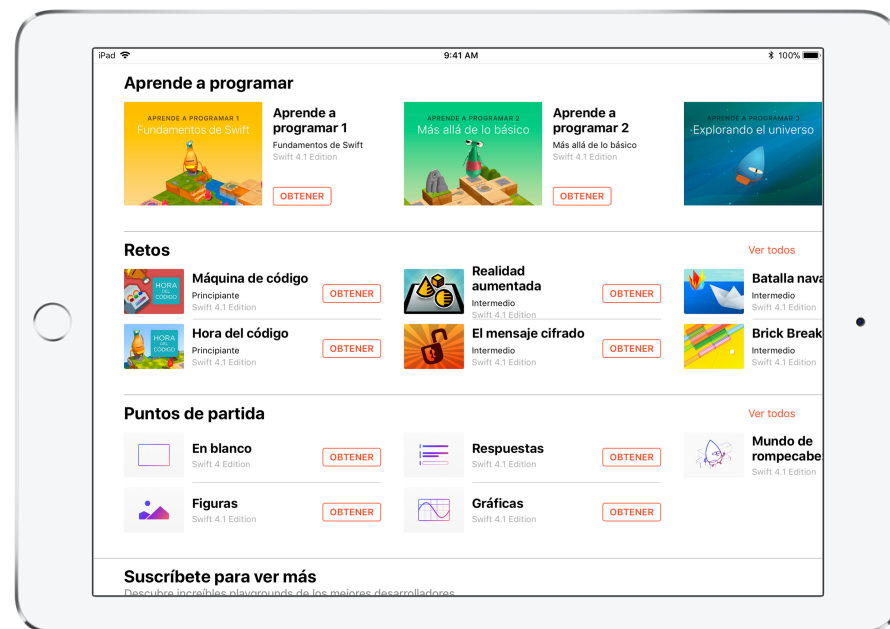
La app incluye un conjunto completo de lecciones diseñadas por Apple, llamado Aprender a programar. Mediante el código Swift real, los estudiantes resuelven juegos de ingenio y conocen personajes que pueden controlar con un simple toque. Al explorar y resolver distintos mundos de juegos de ingenio, los estudiantes desarrollan habilidades de programación que serán la base de sus conocimientos. También se incluyen desafíos adicionales y playgrounds de dispositivos conectados que les permiten a los estudiantes aplicar lo que aprendieron en contextos nuevos.

En Aprender a programar 1 y 2, los estudiantes aprenden conceptos como comandos, depuración, funciones, bucles, algoritmos y otros. Las lecciones no requieren experiencia previa, por lo que son ideales para quienes quieran programar por primera vez. En Aprender a programar 3, los estudiantes amplían sus habilidades de programación para comenzar a pensar como desarrolladores profesionales de apps. Una sección opcional de diseño de apps en la Guía para profesores ayuda a los profesores a guiar a los estudiantes a través de un proceso de diseño de apps.

## En el aula

Aprender a programar 1, 2 y 3, junto con las lecciones de las Guías para profesores, están destinados a estudiantes de escuela secundaria en adelante. El material es flexible y utilizable en cualquier entorno de aprendizaje; además, se puede usar en clases independientes de programación o como parte de un programa de introducción a la programación. Las lecciones están diseñadas para clases de 45 a 60 minutos y algunas abarcan más de una clase. Se incluyen sugerencias del tiempo necesario para completar cada sección de una lección, por lo que, si enseñas una clase menos estructurada, como la de un programa extracurricular, puedes dividir la lección.

Las Guías para profesores incluyen contenido de apoyo para que los profesores puedan enseñar este curso aunque no tengan experiencia en programación. Se recomienda que los estudiantes y profesores tengan un conocimiento básico de los conceptos de programación enseñados en Aprender a programar 1 y 2 antes de avanzar con Aprender a programar 3.



Swift Playgrounds incluye lecciones de programación integradas y desafíos independientes.



# Características principales

**Código Swift: código de iOS real.** En el núcleo de Swift Playgrounds, se encuentra el mismo lenguaje de programación Swift que se usa para crear varias de las apps más importantes que están disponibles en App Store en este momento. Las habilidades que aprenden los estudiantes en Swift Playgrounds no solo se convierten en habilidades útiles en general, sino que son las que necesitarán para crear apps.

**Entorno interactivo.** Crea código en el lado izquierdo de la pantalla y observa inmediatamente los resultados a la derecha, con un solo toque.



**Accesibilidad.** Swift Playgrounds se diseñó teniendo en cuenta la accesibilidad desde el primer momento. Incluye muchas de las potentes funciones de accesibilidad de iOS, como Control por Botón y VoiceOver, e incluso ofrece comentarios de voz adicionales sobre las acciones de los personajes mientras los estudiantes los controlan mediante el código.

## Animaciones impactantes.

Cada sección comienza con una animación impactante que vincula conceptos de programación con la vida real, lo que ayuda a la comprensión de los estudiantes.

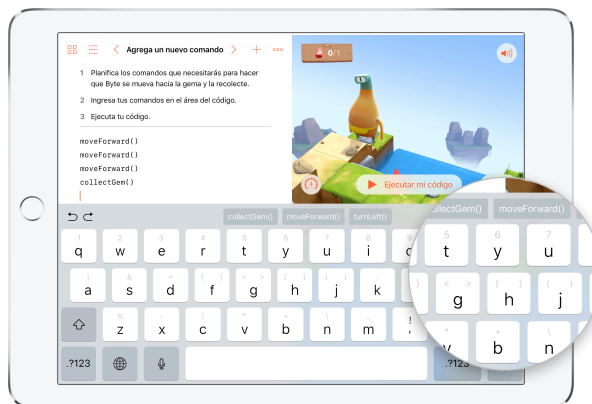
**Glosario integrado.** Las definiciones ayudan a los estudiantes a comprender términos específicos.

## Sugerencias útiles.

Los estudiantes pueden recibir ayuda sobre la marcha si no saben cómo continuar. En muchos casos, las sugerencias son dinámicas y cambian a medida que se ingresa el código.

**Barra de funciones rápidas.** Las sugerencias de código de QuickType aparecen en la parte inferior de la pantalla y permiten que los estudiantes ingresen el código que necesitan con solo tocar la Barra de funciones rápidas.

**Teclado en pantalla.** Un teclado especial para Swift brinda acceso rápido a los números y símbolos que más se usan en Swift.



**Grabar y compartir.** Los estudiantes pueden grabar lo que hacen en pantalla para mostrar su trabajo.

**Revisión del código.** El código se puede ejecutar más rápido o más lento, o puedes detenerte para destacar las líneas a medida que se ejecutan, lo que permite que los estudiantes identifiquen con mayor facilidad dónde pueden ocurrir los errores.

**Tocar para editar.** Arrastra estructuras complejas que encierran otro código, como bucles y definiciones de funciones, alrededor del código existente. Solo tienes que tocar la palabra clave (por ejemplo, "for") y los controles para arrastrar objetos que aparecen en pantalla.



**Editar en el lugar.** Edita valores numéricos, colores y operadores de manera rápida y sencilla con un teclado emergente.

# Recursos de apoyo

## Aprender a programar 1 y 2: Guía para profesores

Esta Guía para profesores, diseñada para utilizarse con estudiantes de escuela secundaria en adelante, ayuda a cualquier profesor a introducir Aprender a programar 1 y 2 en la clase. En las lecciones, se destacan conceptos clave de programación y se demuestra que la programación es una forma de pensar que puede aplicarse a otros temas y a la vida cotidiana. Incluye actividades mejoradas, actividades de revisión y reflexión, una planilla de evaluación y presentaciones de Keynote. La guía está destinada a estudiantes de escuela secundaria en adelante. Representa 40 a 45 horas de lecciones de programación básica, con hasta 45 horas de actividades complementarias que ayudan a los estudiantes a aplicar lo que aprendieron y comenzar a diseñar su propia app. Las equivalencias curriculares se incluyen en las correlaciones con diferentes estándares curriculares nacionales e internacionales en ciencias de la computación.

## Aprender a programar 3: Guía para profesores

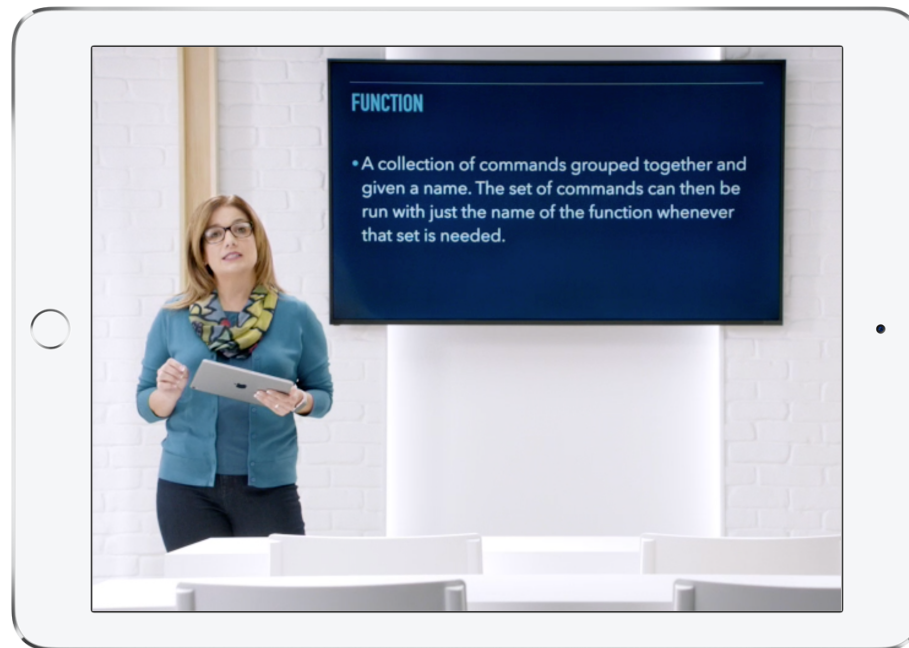
La guía está destinada a estudiantes de escuela secundaria en adelante. Incluye 20 horas de lecciones de programación básica, con hasta 25 horas de actividades complementarias que ayudan a los estudiantes a aplicar lo que aprendieron y comenzar a diseñar su propia app. Partiendo de las habilidades de programación de Aprender a programar 1 y 2, esta guía incluye actividades con historias, lecciones de revisión de código, presentaciones de Keynote, entradas de diario, una planilla de evaluación y más contenido para ayudar a los profesores a introducir estas lecciones en el aula. Las equivalencias curriculares se incluyen en las correlaciones con diferentes estándares curriculares nacionales e internacionales en ciencias de la computación.

## Aprender a programar 1 y 2: Curso de iTunes U

Con este curso de iTunes U, la Guía para profesores de Aprender a programar 1 y 2 cobra vida mediante lecciones en video y recursos adicionales. Además, estos videos sirven para que los profesores vean cómo darles vida a las lecciones de la Guía para profesores en un aula.

## Programa Apple Teacher: Consigue distintivos de Swift Playgrounds

Apple Teacher es un programa de aprendizaje profesional gratuito diseñado para brindar soporte y homenajear a los profesores. Ofrece material de aprendizaje a ritmo personalizado, consejos, inspiración y novedades. Los Apple Teachers pueden visitar Apple Teacher Learning Center para realizar cuestionarios sobre el aprendizaje y la enseñanza con Swift Playgrounds y conseguir cuatro distintivos nuevos. Luego, recibirán un logotipo actualizado de Apple Teacher que presenta Swift Playgrounds para compartir sus logros.



# Resumen del curso

## Aprender a programar 1

Al resolver juegos de ingenio en un mundo dinámico en 3D, los estudiantes desarrollarán un conjunto de habilidades de programación para aumentar su vocabulario básico de programación. La aventura hacia la programación comienza con comandos, funciones y bucles simples. Desde el principio, escribirán código Swift real, el mismo que usan los desarrolladores de verdad.

**Lección 0 | Primeros pasos.** Los estudiantes reciben una introducción a las ciencias de la computación y los objetivos del curso.

**Lección 1 | Pensar como una computadora: Comandos y secuencias.** Los estudiantes aprenden el uso de comandos y secuencias en una situación cotidiana y luego usan comandos y secuencias para programar.

**Lección 2 | Pensar como un detective: Depuración.** Los estudiantes aprenden el uso de la depuración en una situación cotidiana y luego depuran con código.

**Lección 3 | Pensamiento eficiente: Funciones y algo sobre bucles.** Los estudiantes aprenden el uso de las funciones y los bucles "for" en una situación cotidiana, y luego usan funciones y bucles "for" para programar.

**Repaso y reflexión.** Los estudiantes repasan las lecciones 1 a 3, revisan sus carpetas de trabajo y crean una comunidad con revisión entre pares.

**Lección 4 | Pensamiento lógico: Código condicional.** Los estudiantes aprenden el uso de código condicional, booleanos y operadores lógicos, y luego usan código condicional, booleanos y operadores lógicos para programar.

**Lección 5 | Pensar una y otra vez: Bucle "while".** Los estudiantes aprenden el uso de bucles "while" en una situación cotidiana y luego usan bucles "while" para programar.

**Lección 6 | Pensar la misma idea: Algoritmos.** Los estudiantes aprenden el uso de algoritmos en una situación cotidiana y luego usan algoritmos para programar.

**Repaso y reflexión.** Los estudiantes repasan los conceptos de programación de las lecciones 3 a 6, continúan la reflexión en sus carpetas de trabajo y extienden la experiencia en la comunidad.

## Aprender a programar 2

Los estudiantes amplían su conocimiento básico de Swift. Irán más allá de la simple resolución de juegos de ingenio y crearán mundos propios. Aprenderán sobre variables y tipos, los constructos de programación que les permiten almacenar información y acceder a ella. Estas habilidades nuevas, junto con la inicialización y los parámetros, les proporcionarán aún más formas de usar código para interactuar con sus personajes y el mundo del juego, lo que les permitirá cambiar las reglas de ese mismo mundo.

**Lección 7 | Pensar como un Newsbot: Variables.** Los estudiantes aprenden el uso de variables en una situación cotidiana y luego usan variables para programar.

**Lección 8 | Pensar como un arquitecto: Tipos.** Los estudiantes aprenden el uso de tipos en una situación cotidiana y luego usan tipos e inicialización para programar.

**Lección 9 | Pensamiento específico: Parámetros.** Los estudiantes aprenden el uso de parámetros en una situación cotidiana y luego usan parámetros para programar.

**Lección 10 | Pensamiento organizado: Arreglos.** Los estudiantes aprenden el uso de arreglos en una situación cotidiana y luego usan arreglos para programar.

**Proyecto principal.** Los estudiantes crean sus propios mundos usando conceptos que aprendieron durante el programa y crean una historia que se corresponda con ese mundo.

Los estudiantes reflexionan sobre lo que aprendieron mediante sus carpetas de trabajo y la revisión entre pares de la comunidad.

**Diseño de apps.** Los estudiantes completan las etapas de un ciclo de diseño que se centra en la preparación de un prototipo, un proceso similar al que realizan los desarrolladores de apps profesionales.



# Resumen del curso (continuación)

## Aprender a programar 3

En Aprender a programar 3, los estudiantes amplían las habilidades de programación que aprendieron en las lecciones anteriores para comenzar a pensar como un desarrollador profesional de apps. Aprender a programar 1 y 2 es un requisito previo recomendado para Aprender a programar 3.

En el espacio interestelar del universo de Blu, los estudiantes crean un conjunto de herramientas creativas mientras aprenden conceptos útiles de programación que usan los desarrolladores profesionales. A medida que aprendan sobre gráficos y coordenadas, podrán colocar y manipular imágenes, y luego combinar estas técnicas con eventos táctiles para pintar formas artísticas en el espacio.

Después de aprender sobre los eventos táctiles, los estudiantes aprenderán sobre las cadenas y adquirirán una nueva herramienta para darle voz al silencioso universo de Blu. Finalmente, explorarán los controladores de eventos a medida que usen eventos reales, como movimientos de dedos o toques, para activar su código. Con los controladores de eventos, crearán extraterrestres animados o convertirán el universo en un instrumento musical gigante. Cuando terminen, habrán combinado sus habilidades con total dominio y habrán escrito su código más avanzado hasta el momento.

### Lección 1 | Introducción a Aprender a programar 3: Coordenadas.

Los estudiantes aprenden sobre las coordenadas, repasan algoritmos, bucles "for" y arreglos, y luego combinan conceptos para programar. También debaten sobre las características que debe tener una historia visual excelente.

### Lección 2 | Pensar como un diseñador de apps: Eventos táctiles.

Los estudiantes repasan variables, tipos e inicialización mediante el análisis de sus apps favoritas. Luego, crean e inicializan sus propias herramientas de imagen en Swift Playgrounds. También investigan cómo influyen las imágenes en las historias visuales.

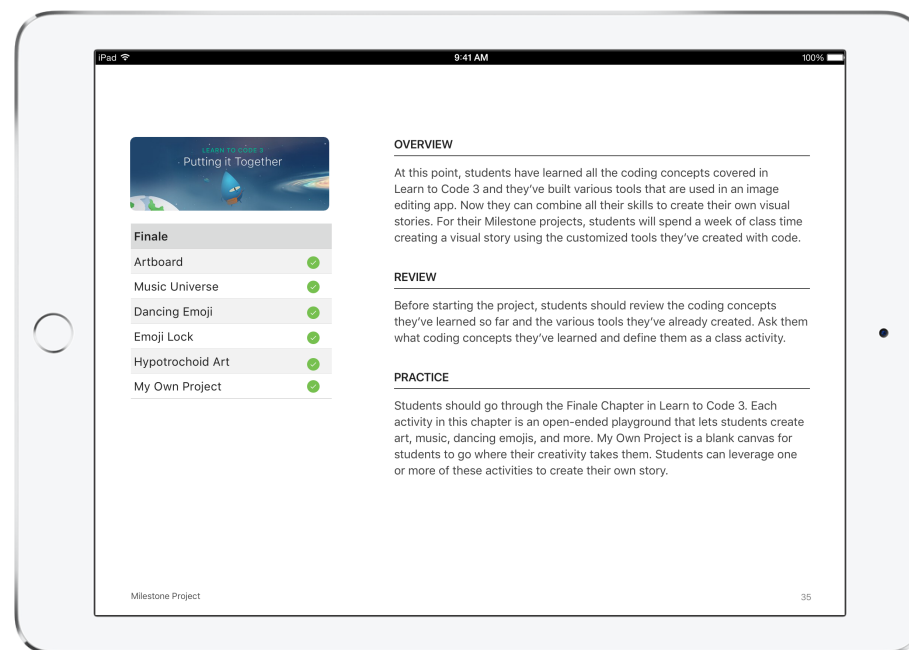
**Lección 3 | Pensar como un editor: Cadenas.** Los estudiantes aprenden sobre las cadenas en situaciones cotidianas y luego crean sus propias herramientas de texto en Swift Playgrounds. También investigan cómo influye el texto en las historias visuales.

### Lección 4 | Pensar como un animador: Controladores de eventos.

Los estudiantes aprenden sobre los controladores de eventos diseñando sus propios juegos. Luego, crean sus propias herramientas de acción en Swift Playgrounds e investigan cómo influye la interactividad en las historias visuales.

**Proyecto principal.** Los estudiantes programan sus propias historias visuales en Swift Playgrounds.

**Diseño de apps.** Los estudiantes completan las etapas de un ciclo de diseño que se centra en la preparación de un prototipo, un proceso similar al que realizan los desarrolladores de apps profesionales.



# Información adicional

## Swift Playgrounds requiere iOS 10 y funciona en los siguientes modelos de iPad:

- iPad Pro (9,7 pulgadas)
- iPad Pro (12,9 pulgadas)
- iPad
- iPad Air 2
- iPad Air
- iPad mini 4
- iPad mini 3
- iPad mini 2

## Descargar los recursos de Swift Playgrounds

- [Aprender a programar 1 y 2: Curso de iTunes U](#)
- [Aprender a programar 1 y 2: Guía para profesores](#)
- [Aprender a programar 3: Guía para profesores](#)
- [App Swift Playgrounds](#)

## Descargar las guías de Desarrollo de apps con Swift

- [Introducción al desarrollo de apps con Swift](#)
- [Introducción al desarrollo de apps con Swift: Guía para profesores](#)
- [Desarrollo de apps con Swift](#)
- [Desarrollo de apps con Swift: Guía para profesores](#)

## Descargar los recursos de Empezar a programar

- [Tynker](#)
- [codeSpark Academy](#)
- [Empezar a programar 1](#)
- [Empezar a programar 2](#)

## Recursos adicionales

- Obtén más información sobre [Swift Playgrounds](#).
- Obtén más información sobre el programa [Programación para todos](#).
- Obtén más información sobre [Swift](#).
- Conéctate con otros docentes en los [foros de Apple Developer](#).

# Correlación curricular: Aprender a programar 1, 2 y 3

Esta es la correlación preliminar entre las Guías para profesores de Aprender a programar 1, 2 y 3 Swift Playgrounds y los estándares provisionales de 2016 para las ciencias de la computación K–12 de nivel 2 de la Asociación de Profesores de Ciencias de la Computación (CSTA). La correlación abarca los conceptos de algoritmos y programación en los estándares provisionales para las ciencias de la computación K–12 de 2016 de la CSTA.

Estándares para las ciencias de la computación K-12 de la CSTA, nivel 2 (grados 6 a 8)										
Estándar de la CSTA	2-A-2-1 Devolución de los pares	2-A-7-2 Algoritmos: Comparación	2-A-7-3 Atribución	2-A-7-4 Algoritmos: Interpretación	2-A-5-5 Artefactos informáticos	2-A-5-6 Desarrollo de programas	2-A-5-7 Variables	2-A-4-8 Parámetros	2-A-3-9 Descomposición	2-A-6-10 Diseño iterativo
Capítulos de Swift Playgrounds	Comandos	●	●						●	●
	Funciones	●	●		●	●		●	●	●
	Bucles "for"	●	●		●	●		●	●	●
	Código condicional	●	●		●	●		●	●	●
	Operadores lógicos	●	●		●	●		●	●	
	Bucles "while"	●	●		●	●		●	●	●
	Algoritmos	●	●		●	●		●	●	●
	Variables	●	●		●	●	●	●	●	●
	Tipos	●	●		●	●	●	●	●	●
	Inicialización	●	●					●	●	
	Parámetros	●	●		●	●		●	●	
	Creación del mundo	●	●		●	●		●	●	
	Arreglos	●	●		●	●	●	●	●	●
	Coordenadas	●	●		●	●	●	●	●	●
	Eventos táctiles	●	●	●	●	●	●	●	●	●
	Controladores de eventos	●	●	●	●	●	●	●	●	●
	Final	●	●		●	●	●	●	●	●

Clave: ● Se correlaciona con el estándar



# Correlación curricular: Aprender a programar 1 y 2

Esta es la correlación preliminar entre Aprender a programar 1 y 2: Guía para profesores de Swift Playgrounds y los estándares provisionales de 2016 para las ciencias de la computación K–12 de nivel 2 de la Asociación de Profesores de Ciencias de la Computación (CSTA). La correlación abarca los conceptos de algoritmos y programación en los estándares provisionales para las ciencias de la computación K–12 de 2016 de la CSTA.

Estándares para las ciencias de la computación K-12 de la CSTA, nivel 2 (grados 6 a 8)										
Estándar de la CSTA	2-A-2-1 Devolución de los pares	2-A-7-2 Algoritmos: Comparación	2-A-7-3 Atribución	2-A-7-4 Algoritmos: Interpretación	2-A-5-5 Artefactos informáticos	2-A-5-6 Desarrollo de programas	2-A-5-7 Variables	2-A-4-8 Parámetros	2-A-3-9 Descomposición	2-A-6-10 Diseño iterativo
Capítulos de Swift Playgrounds	Comandos	●	●		●				●	●
	Funciones	●	●		●	●		●	●	●
	Bucles "for"	●	●		●	●		●	●	●
	Código condicional	●	●		●	●		●	●	●
	Operadores lógicos	●	●		●	●		●	●	
	Bucles "while"	●	●		●	●		●	●	●
	Algoritmos	●	●		●	●		●	●	●
	Variables	●	●		●	●	●	●	●	●
	Tipos	●	●		●	●	●	●	●	●
	Inicialización	●	●					●	●	
	Parámetros	●	●		●	●		●	●	
	Creación del mundo	●	●		●	●		●	●	
	Arreglos	●	●		●	●	●	●	●	●

Clave: ● Se correlaciona con el estándar

# Correlación curricular: Aprender a programar 3

Esta es la correlación preliminar entre Aprender a programar 3: Guía para profesores de Swift Playgrounds y los estándares provisionales de 2016 para las ciencias de la computación K–12 de nivel 2 de la Asociación de Profesores de Ciencias de la Computación (CSTA). La correlación abarca los conceptos de algoritmos y programación en los estándares provisionales para las ciencias de la computación K–12 de 2016 de la CSTA.

Estándares para las ciencias de la computación K-12 de la CSTA, nivel 2 (grados 6 a 8)											
Capítulos de Swift Playgrounds	Estándar de la CSTA	2-A-2-1 Devolución de los pares	2-A-7-2 Algoritmos: Comparación	2-A-7-3 Atribución	2-A-7-4 Algoritmos: Interpretación	2-A-5-5 Artefactos informáticos	2-A-5-6 Desarrollo de programas	2-A-5-7 Variables	2-A-4-8 Parámetros	2-A-3-9 Descomposición	2-A-6-10 Diseño iterativo
	Coordenadas	●	●		●	●	●	●	●	●	●
	Eventos táctiles	●	●	●	●	●	●	●	●	●	●
	Controladores de eventos	●	●	●	●	●	●	●	●	●	●
	Final	●	●		●	●	●	●	●	●	●

Clave: ● Se correlaciona con el estándar

Las funcionalidades están sujetas a cambios. Es posible que algunas funcionalidades no estén disponibles en todas las regiones o en todos los idiomas.