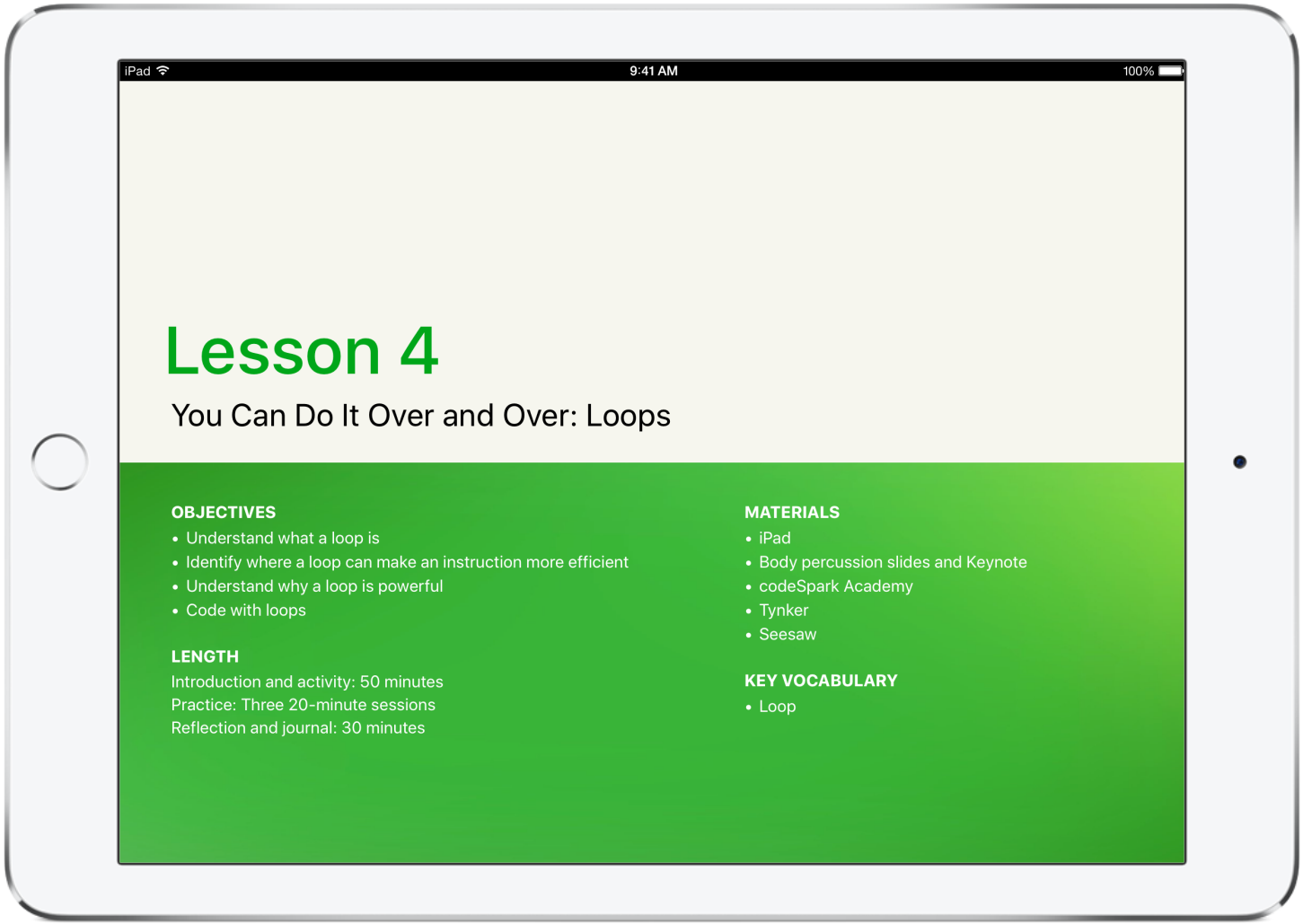
Guía de «Get Started with Code»

Septiembre de 2017

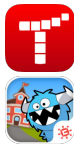
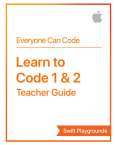
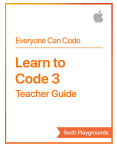
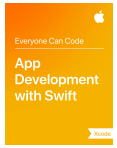


Programación para todos

La tecnología tiene su propio lenguaje: la programación. Nosotros creemos que aprender a programar es una habilidad básica. Te enseña a resolver problemas, a trabajar en equipo de formas creativas y a diseñar apps que dan forma a tus ideas. Creemos que todas las personas deberían tener la oportunidad de crear algo capaz de cambiar el mundo. Por eso hemos diseñado un programa con todas las herramientas y recursos que permiten a todo el mundo aprender y enseñar programación.

Plan de estudios de «Programación para todos»

El programa «Programación para todos» incluye distintos recursos que guiarán a los estudiantes durante el proceso de creación de apps aunque no tengan experiencia. En la tabla de abajo puedes ver un resumen de los recursos de enseñanza y aprendizaje gratuitos que están disponibles.

Plan de estudios	Dispositivo	Público objetivo	App	Requisitos	Presentación	Materiales didácticos	Recursos de apoyo	Número de horas
		Desde infantil hasta 2.º de primaria		Ninguno	Empezar a pensar como programadores mediante la exploración de conceptos con apps visuales.	• Lecciones de la app codeSpark Academy • Curso Cadete Espacial de Tynker	• «Get Started with Code 1» (guía para profesores)	30 horas en las que se incluyen la guía para profesores y lecciones de apps
		De 3.º a 5.º de primaria		Ninguno	Explorar conceptos básicos de programación y empezar a pensar como un programador con apps visuales.	• Curso Hechizos de Dragón de Tynker	• «Get Started with Code 2» (guía para profesores)	36 horas en las que se incluyen la guía para profesores y lecciones de apps
		A partir de secundaria		Ninguno	Aprender conceptos básicos de programación con código de Swift.	• App Swift Playgrounds • Lecciones de «Aprende a programar 1 y 2» • Curso de iTunes U	• «Aprender a programar 1 y 2» (guía para profesores) • Distintivos de Swift Playgrounds del Apple Teacher Learning Center	Hasta 85 horas en las que se incluyen la guía para profesores y las lecciones de «Aprende a programar 1 y 2»
		A partir de secundaria		Aprende a programar 1 y 2	Ampliar las habilidades de programación y pensar como un desarrollador de apps.	• App Swift Playgrounds • Lecciones de «Learn to Code 3»	• «Learn to Code 3» (guía para profesores)	Hasta 45 horas en las que se incluyen la guía para profesores y las lecciones de «Learn to Code 3»
		Secundaria y universidad		Ninguno	Adquirir experiencia con las herramientas, las técnicas y los conceptos necesarios para crear una app básica para iOS desde cero.	Libro «Introducción al desarrollo de apps con Swift» y archivos del proyecto	• «Introducción al desarrollo de apps con Swift» (guía para profesores)	90 horas
		Secundaria y universidad		Ninguno	Conocer más a fondo Swift, UIKit y las redes mediante talleres prácticos y proyectos guiados. Los estudiantes serán capaces de diseñar su propia app al terminar del curso.	Libro «App Development with Swift» y archivos del proyecto	• «App Development with Swift» (guía para profesores)	180 horas

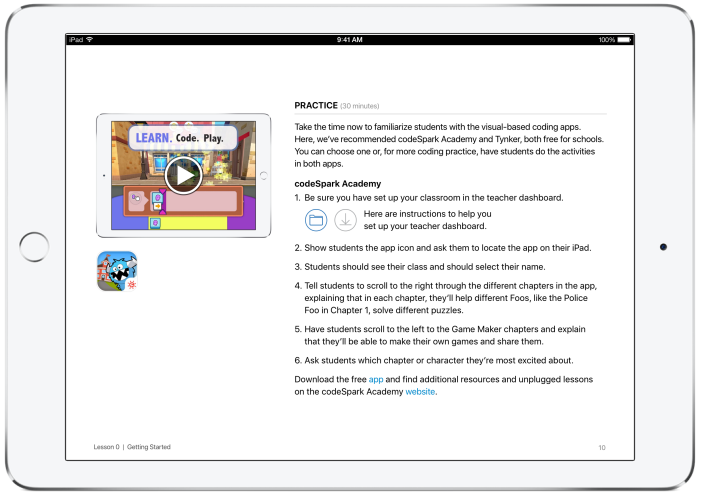
Descripción

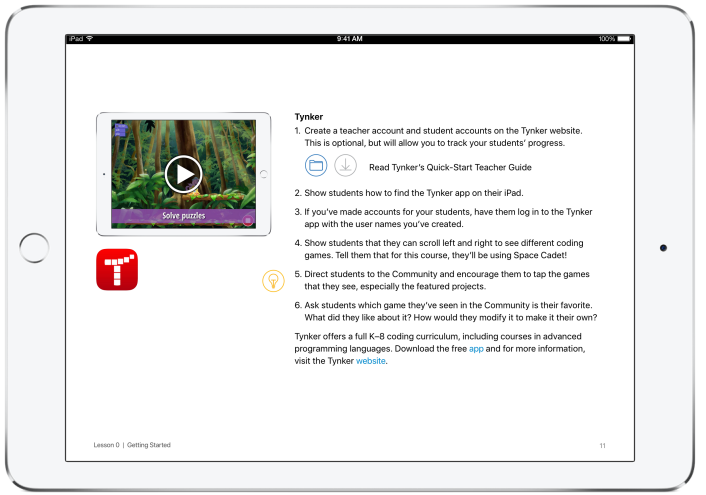
Los primeros años de escolaridad son un momento propicio para introducir ciertas nociones de programación para pensar acerca del mundo cotidiano y digital, así como para desarrollar las bases fundacionales del pensamiento computacional. Las apps diseñadas para enseñar a los más jóvenes, como codeSpark Academy y Tynker, utilizan rompecabezas visuales para ayudar a desarrollar las habilidades de solución de problemas y fomentar tanto la perseverancia como la creatividad. codeSpark Academy está dirigida a estudiantes de cinco a siete años. Su interfaz sin texto permite jugar a niños que aún no han aprendido a leer, no saben inglés o tienen dificultades de lectura. En Tynker, los niños de entre cinco y once años pueden experimentar con bloques visuales y pasar a la programación con texto a medida que resuelven rompecabezas y hacen proyectos.

En el aula

Tynker y codeSpark Academy, junto con las lecciones de las guías para profesores de «Get Started with Code», te ayudan a introducir la programación en los primeros cursos de primaria. Cada lección está dedicada a un concepto de programación básico y demuestra que la programación es una forma de pensar que puede aplicarse tanto a otros campos del conocimiento como a la vida cotidiana en general.

Las guías para profesores te dan todo el apoyo que necesitas para ayudar a los estudiantes a resolver los rompecabezas de programación sea cual sea su nivel. También se incluyen actividades de ampliación y de diseño de apps, preguntas para reflexionar, sugerencias para escribir un diario, una rúbrica y otros recursos que te ayudarán a reforzar la comprensión de los estudiantes. Las lecciones pueden impartirse como un bloque o por secciones. Los mapas de correlaciones en los apéndices ofrecen ofrece una correlación preliminar de las lecciones con los estándares Computer Science Teachers Association K–12 y Computer Science Standards for Level 1.





Características

codeSpark Academy

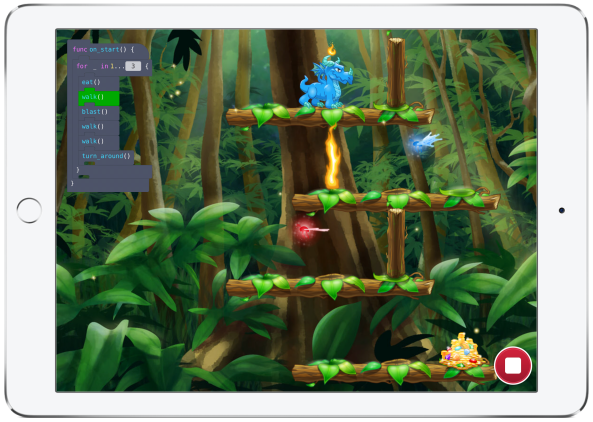
Aprendizaje. Los estudiantes resuelven rompecabezas en este juego sin texto para aprender conceptos básicos de la informática como las secuencias, los ciclos, las sentencias condicionales y mucho más.

Creación. Los estudiantes ponen en práctica lo aprendido programando sus propios proyectos en Fabricante de Juegos.

Autoguiado. No hace falta saber programación para aprender, enseñar o jugar. El plan de estudios y el tablero del profesor, que muestra informes de progreso, están disponibles de forma gratuita

Tynker

Entorno de programación. Los estudiantes resuelven rompecabezas estructurados a su propio ritmo para aprender conceptos y aplicarlos de forma creativa.



Evaluación automática. Un tablero del profesor evalúa las habilidades de los estudiantes mediante rompecabezas, cuestionarios y análisis de código.

Integración con Swift. A medida que los estudiantes resuelven rompecabezas, pueden cambiar entre bloques visuales y bloques de Swift para familiarizarse con este lenguaje de cara a futuros proyectos.

Guías «Get Started with Code» para profesores

Archivos descargables. Los archivos de las plantillas para actividades y las presentaciones de Keynote facilitan el aprendizaje de los estudiantes en clase.

Claves de respuestas. Las soluciones de los rompecabezas de Tynker y codeSpark Academy te lo ponen fácil para ayudar a los estudiantes que no consiguen avanzar.

Trabajos de alumnos de ejemplo. Echa un vistazo a los posibles resultados de las actividades.

Reflexiones. Estas preguntas diseñadas para los debates en clase te ayudarán a repasar y reforzar la conexión entre aplicar un concepto dentro de un entorno de programación y hacerlo fuera de él.

Consejos y ejemplos. Se incluyen ideas de todo tipo para ampliar o simplificar las lecciones.

Actividades de diseño de apps. Estas lecciones guían a los estudiantes por un proceso de diseño que permite crear un concepto y un prototipo para una app que resuelva un problema de la clase o el centro.

Resumen del curso

Get Started with Code 1

Los estudiantes estudian conceptos de programación de forma interactiva, práctica y aplicados a situaciones cotidianas y empiezan a pensar como programadores. Descubrirán cosas como los comandos, las secuencias, los ciclos, los eventos y los algoritmos. También colaborarán para predecir el resultado del código, depurarán tanto el suyo como el de sus compañeros y pondrán en práctica sus habilidades utilizando apps de programación visual, resolviendo rompecabezas y diseñando sus propios proyectos. Las actividades opcionales de diseño guían a los estudiantes por un proceso de diseño que permite crear un concepto y un prototipo para una app que resuelva un problema de la clase o el centro.

Lesson 0—Getting Started. Descubre lo que los estudiantes saben sobre apps y programación, organiza el muro de trabajo de la clase y habla de las apps principales que usarán en las lecciones. Los estudiantes también aprenderán cuáles son las distintas funciones en un equipo de diseño de apps.

Lesson 1—You Can Order It: Introduction to Sequencing. Los estudiantes verán secuencias habituales, crearán una partiendo de una situación real y resolverán rompecabezas en apps de programación visual utilizando secuencias de ejemplo. También descubrirán las distintas funciones que realizan las apps.

Lesson 2—You Can Step It: Creating Sequences. Al examinar la importancia del orden a la hora de crear secuencias de instrucciones, los estudiantes aprenderán que las mismas acciones se pueden intercambiar para crear otras secuencias y diseñar su propia coreografía. También descubrirán nuevos comandos, resolverán problemas más complejos en las apps de programación y compararán distintas apps que ayudan a los usuarios a aprender nuevas ideas.

Lesson 3—You Can Choose It: Flexible Sequencing. Los estudiantes descubrirán que se puede cambiar el orden de los pasos de una secuencia, crearán secuencias flexibles, verán las distintas formas en que se puede resolver un rompecabezas y compartirán sus soluciones con los demás. También conocerán apps que ayudan a los usuarios a comunicarse con otras personas de su entorno.

Lesson 4—You Can Do It Over and Over: Loops. Los estudiantes identificarán ciclos en contextos cotidianos, utilizarán la creatividad para crear sus propios ciclos de percusión corporal, verán cómo se representan en lenguaje de programación y los utilizarán para simplificar su código. También aprenderán a diseñar una interfaz y a hacer que una app sea intuitiva y fácil de usar.

Lesson 5—You Can Fix It: Debugging. Los estudiantes descubrirán la importancia de la depuración en distintos contextos y aplicarán sus nuevas habilidades para superar un reto y depurar las soluciones de sus compañeros. También practicarán prediciendo la salida de su código, identificarán errores que evitan que se ejecute como estaba previsto y aprenderán diseñar apps para solucionar problemas.

Lesson 6—You Can Prompt It: Events and Actions. Los estudiantes descubrirán cómo los eventos pueden hacer que su app y su código funcionen mejor, aprenderán a programar con eventos, verán cómo accionamos eventos en el día a día y crearán un mando a distancia para un robot que les ayudará a practicar con los eventos de llamada. También empezarán a diseñar sus propia apps.

Lesson 7—You Can, If You Follow the Rule: IF Statements. Los estudiantes descubrirán las sentencias IF y aprenderán a identificarlas en las cosas de todos los días. Por ejemplo, en las reglas de los juegos de mesa que conocen. Empezarán a programar con sentencias IF, harán que su código responda mejor a las condiciones del entorno y utilizarán diagramas de flujo para enseñar cómo funcionan sus apps.

Lesson 8—You Can Solve It: Algorithms. Los estudiantes pondrán en práctica todo lo aprendido diseñando algoritmos que sigan una serie de pasos para resolver un problema. Empezarán por lo básico e irán avanzando hasta crear un laberinto. También harán prototipos de sus apps y realizarán un proyecto final en el que tendrán que documentar el proceso de diseño.

Resumen del curso (continuación)

Get Started with Code 2

En «Get Started with Code 2», los estudiantes aprenderán conceptos básicos de programación y empezarán a pensar como programadores. Verán los algoritmos, las funciones, los ciclos, las sentencias condicionales, las variables y los aspectos básicos del diseño de interfaces. Trabajarán en grupos y de forma individual mientras mejoran sus habilidades resolviendo problemas reales, probando el código de sus compañeros y diseñando programas para bots. También practicarán estas habilidades en Tynker, donde tendrán que resolver problemas aplicando los conceptos aprendidos en las actividades de clase. Las actividades opcionales de diseño guían a los estudiantes por un proceso de diseño que permite crear un concepto y un prototipo para una app que resuelva un problema de la clase o el centro.

Lesson 0—Getting Started. Descubre lo que los estudiantes saben sobre apps y programación, haz que se familiaricen con una app de programación visual como Tynker y organiza sus diarios digitales con una app como Seesaw. También se enfrentarán a un reto relacionado con el diseño de apps.

Lesson 1—Think in Steps: Solving Problems with Algorithms. Los estudiantes conocerán los algoritmos, que son conjuntos de instrucciones para resolver un problema o realizar una tarea. Incubarán un dragón en la app Tynker y crearán un algoritmo para resolver rompecabezas mientras aprenden a utilizar las secuencias. También diseñarán y probarán algoritmos en el aula y harán lluvias de ideas para crear apps con las que resolver problemas.

Lesson 2—Think in Fixes: Debugging. Los estudiantes aprenderán a encontrar y arreglar errores en su algoritmo y su código. Tendrán que modificar algoritmos en Tynker con el objetivo de crear un programa que resuelva los rompecabezas correctamente. También se familiarizarán con el papel de los teclados en las apps y verán cómo aplicarlos a sus propias ideas.

Lesson 3—Think in Circles: Looking for Loops. Los alumnos descubrirán los ciclos, que sirven para repetir las mismas acciones, y diseñarán y probarán un algoritmo para controlar una serpiente. Utilizarán ciclos for en Tynker para resolver rompecabezas mediante la identificación de patrones y harán lluvias de ideas sobre cómo pueden usar la cámara integrada y el micrófono en su app.

Lesson 4—Think in Bits: Composition and Decomposition. Para crear un algoritmo para la canción del vaso, los estudiantes dividirán la rutina en distintos movimientos. En Tynker, resolverán problemas dividiéndolos en problemas más pequeños. Tendrán que pensar cómo usar la pantalla táctil para que su app sea más interactiva.

Lesson 5—Think in Sets: Abstraction. Los estudiantes descubrirán los conceptos de similitud y generalización mientras categorizan objetos en grupos y explican su lógica. En Tynker, usarán la abstracción para descubrir similitudes entre problemas y resolverán rompecabezas cada vez más complejos con sus nuevas herramientas de programación. También pensarán en cómo usar el Bluetooth y otras herramientas para conectarse a dispositivos cercanos.

Lesson 6—Think in Patterns: Forming Functions. Los estudiantes tendrán que crear una rutina para un robot de comandos, dividirla en funciones e intercambiar algoritmos para comprobar si los resultados son los esperados. En Tynker, los estudiantes utilizarán nombres y funciones de llamada mientras reutilizan instrucciones para programar de una forma más eficiente. También aprenderán a qué tipos de datos se puede conectar su app mediante GPS.

Lesson 7—Think in Specifics: Conditional Statements. Los estudiantes se embarcarán en una aventura virtual en la que tendrán que definir su destino con una serie de condiciones expresadas mediante sentencias. En Tynker, utilizarán las sentencias para gestionar las decisiones y las alternativas en los rompecabezas. También harán lluvias de ideas para lograr que sus apps sean únicas.

Resumen del curso (continuación)

Lesson 8—Think in Cycles: While Loops and Nested Loops. Los estudiantes utilizarán ciclos while y ciclos anidados para lograr que su robot escarche suficientes rosquillas para cada cliente. En Tynker, los estudiantes usarán algoritmos para acortar su código. También formarán equipos de diseño y empezarán a crear el prototipo de su app.

Lesson 9—Think In and Outside the Box: Variables, Input, and Output. Los estudiantes tendrán que utilizar variables para diseñar un algoritmo que les permita declamar un poema o cantar una canción de rap adaptado a su público. En Tynker, utilizarán variables a medida que los rompecabezas vayan subiendo de dificultad y pondrán en práctica todo lo aprendido hasta el momento. También tendrán que hacer entrevistas para saber cuál es el público objetivo de su app.

Lesson 10—Think in Practice: Design UI. Los estudiantes analizarán en qué consiste un buen diseño y crearán un cartel para su centro. En Tynker, tendrán que utilizar todas sus habilidades y completar las lecciones de «Get Started with Code 2». Los estudiantes aprenderán sobre la interfaz y la experiencia del usuario y crearán un tablero de inspiración para el diseño de su app. Y en el proyecto final, crearán una presentación para su app.

Información adicional

Recursos de «Get Started with Code»

- Tynker
- codeSpark Academy
- Get Started with Code 1
- Get Started with Code 2

Recursos de Swift Playgrounds

- Learn to Code 1 & 2 (curso de iTunes U)
- Aprender a programar 1 y 2 (guía para profesores)
- Learn to Code 3 (guía para profesores)
- App Swift Playgrounds

Guías de «App Development with Swift»

- Introducción al desarrollo de apps con Swift
- Guía de «Introducción al desarrollo de apps con Swift» para profesores
- App Development with Swift
- Guía de «App Development with Swift» para profesores

Otros recursos

- Más información sobre el programa Programación para todos.
- Más información sobre Swift.
- Más información sobre Xcode.
- Conecta con otros educadores en los foros de Apple Developer.
- Más información sobre codeSpark Academy.
- Más información sobre Tynker.

Las prestaciones están sujetas a cambios. Algunas prestaciones no están disponibles en todos los países o idiomas.

© 2017 Apple Inc. Todos los derechos reservados. Apple, el logotipo de Apple, iTunes U, Keynote y Xcode son marcas comerciales de Apple Inc., registradas en EE. UU. y en otros países. Swift y Swift Playgrounds son marcas comerciales de Apple Inc. Los demás nombres de productos y empresas mencionados en este documento son marcas comerciales de sus respectivas empresas.
Las especificaciones de producto están sujetas a cambios sin previo aviso. Este documento se proporciona con fines meramente informativos; Apple no asume ninguna responsabilidad relacionada con su uso. Septiembre de 2017