



DESCUBRIENDO
SHORTCUTS

PARA iOS

Descubriendo Shortcuts para iOS

Frank Liñán García

Rafael Antonio Roa Hernández

Jose Ruiz Gutierrez

2019-01-02

Título original: Descubriendo Shortcuts para iOS

1ª Edición: Enero de 2019.

©Frank Liñán García; Rafael Antonio Roa Hernández;
Jose Ruiz Gutierrez

ISBN: 978-84-09-07977-3

<http://www.shrtcts.es>

Todos los derechos reservados. Queda prohibido, sin autorización escrita de los titulares del copyright la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la reprografía o el tratamiento informático.

Agradecimientos

Dedicado a Maite (sin ti no hubiera sido posible), Xènia ('T'estimo infinit') y Aleix. A mi madre Anna y a mi padre que lo estará viendo desde allá arriba. A todos los que me habéis hecho mejor persona, gracias por soportarme.

Frank

A Inma, la mujer de mi vida y a Daniel, mi futuro hijo, por las charlas que tendremos. A mis padres y mis hermanas, por quererme como soy.

Rafa

A mi madre por ser, a mi padre por seguir, a mi hermana por estar, y a Paloma por existir.

Jose

Novedades v1.1

- Añadido índice
- Corrección de errores

Novedades v2.0

- Comentadas las acciones de la lista del capítulo 3.2 Acciones por categoría
- Actualizado apartado 3.4 Dedicado a las URL Schemes.
- Añadido apartado 7.1.2 - Listas más enriquecidas con vCard
- Añadido Capítulo 11: Shortcuts y Salud
- Añadidos 30 ejemplos documentados en 12.2: Ejemplos útiles
- Añadido ejemplo en 12.2.3. Copia de seguridad de tus códigos de homekit
- Corrección de errores

Novedades v2.1

- Añadidas las nuevas acciones de Shortcuts 2.2 en el capítulo 3.2 Acciones por categoría
- Añadido ejemplo con las nuevas acciones de Notas en el capítulo 12.2

Novedades v2.2

- Añadido 3.4.4 Ejemplo X-CallbackURL completo mostrando un ejemplo con Bear App para iOS
- Añadido apartado 5.5.1 JAYSON
- Añadido apartado 7.4.4 donde se explica con ejemplos cómo actualizar valores en diccionarios anidados

Índice

Agradecimientos

Prólogo

Los autores

1. Introducción

1.1. Historia de Shortcuts

1.2. Programación visual

2. Conceptos básicos

2.1. Identificar Tareas Repetitivas

2.2. shortcuts vs Siri shortcuts.

2.3. Interfaz y funcionamiento

2.4. Como funciona Shortcuts

3. Acciones

3.1. Concepto de Entrada - Salida

3.2. Acciones por categoría

3.3 Tipos de acciones

3.4 Relación de Shortcuts con otras Apps: URL-Schemes

3.5 Comentarios

4. Variables

4.1. Shortcuts y las variables

4.2. Definir, Obtener y Añadir

4.3. Variables Mágicas

4.4. Variables Especiales

4.5. Vida de una variable

5. Otros conceptos útiles

- 5.1. Markdown
- 5.2. RegEx para principiantes
 - 5.2.2. Grupos
 - 5.2.3. Coincidencias Greedy y Lazy
 - 5.2.4. LookAhead y LookBehind
 - 5.2.5. Ejemplos
- 5.3. Capturar datos de una web abierta
- 5.4. Fechas
- 5.5. JSON
- 5.6. Base64

6. Trabajando con archivos

- 6.1 Trabajando con archivos multimedia
- 6.2 Trabajando con cualquier tipo de archivos

7. Conceptos avanzados

- 7.1 Listas
- 7.2. Flujos de ejecución
- 7.3. Reutilización de código en atajos
- 7.4. Diccionarios
- 7.5 Estructuras de control

8. Web APIs

- 8.1. Qué es una API
- 8.2. Web Services: SOAP APIs y REST APIs
- 8.3. Recursos
- 8.4. Otros Aspectos de las REST APIs
- 8.5. Resumen y Ejemplos

9. Como ejecutar los shortcuts

9.1. Recordatorios

9.2. IFTTT

9.3. Launch Center Pro / Launcher

9.4. DUE

10. Puliendo y optimizando shortcuts

10.1. Compartir y preguntas de importación

10.2. Debugging: Herramientas de ayuda

11. Shortcuts y Salud

12. Ejemplos prácticos

12.1. Para aprender

12.2. Ejemplos útiles

Índice de ejemplos

Prólogo

Por Oliver Nabani

Nuestra relación con la tecnología está cambiando continuamente. A lo largo del tiempo nos ha sucedido que lo que eran simples herramientas, se han convertido en una parte de nuestra vida más relacionada con la moda que con el uso de esa tecnología como tal.

Todos nos hemos decantado por un coche u otro dependiendo de su estética o su equipamiento, más allá de la hoja de especificaciones técnicas que define al propio vehículo. Muchos hemos pasado de meter en casa esos ordenadores cuadrados de color beige a otros con diseños de tal valor que se exhiben orgullosamente en el MoMA de Nueva York. Hemos visto tendencias y modas en lo que a materiales y colores se refiere: del plástico al aluminio o al cristal, del beige al acero o el oro rosa... En definitiva han pasado de ser simples herramientas de trabajo a formar parte de nuestro día a día, donde nos acompañan, nos ayudan, e incluso dependiendo de su diseño, nos visten.

Y es que hay pocas cosas que nos definen tan bien a los seres humanos como es el concepto de la personalización. Nos encanta sentirnos diferentes y tener cosas únicas, exclusivas. Nos encanta tener objetos que nos ayuden a diferenciarnos como individuos.

Sin embargo los conceptos de unicidad y exclusividad chocan frontalmente con los de economía de escala y globalización. Para que una tecnología sea rentable y pueda desarrollarse a un ritmo adecuado, debe producirse a gran escala. Las empresas necesitan vendernos millones de dispositivos iguales para poder repartir el coste de su desarrollo y poder garantizar futuras investigaciones.

Para llegar a un equilibrio entre exclusividad y globalización, las soluciones se adaptan al uso de la tecnología en sí: en un coche puedes personalizar desde su estética a su equipamiento, en un ordenador puedes decidir qué software instalar o qué periféricos usar.

Si nos adentramos en el mundo de los Smartphones, mundo que debe interesarle si ha escogido este libro, vemos que esta tendencia es clara. La personalización se ha convertido en el caballo de batalla de muchas marcas para buscar esa diferenciación en el mercado. Yo mismo he presenciado, e incluso participado, en esos debates sin fin sobre qué plataforma (Android o iOS) es mejor en cuanto a experiencia de uso. Y uno de los pilares fundamentales de la defensa “androide” es la capa de personalización que permite este sistema: usar tus propios iconos, colores, animaciones, gestos... en muchos casos son motivo suficiente para elegir un sistema por encima de su rival.

Sin embargo, parece evidente que esta parte de la personalización es solo una cara de la moneda. El otro elemento que nos va a permitir diferenciar nuestro Smartphone del de los demás son las Apps que tengamos instaladas: el equivalente al equipamiento de un coche.

Hay, literalmente, millones de Apps. Prácticamente hay una para cada idea que se nos pase por la cabeza. Es un negocio multimillonario que intenta cubrir todas nuestras necesidades pero, como todo negocio globalizado, no puede llegar al grado de personalización que los usuarios querríamos.

Para alguien inquieto como yo, la solución a este problema pasó por algo impensable para la mayoría: adquirir el kit de desarrollo de Apple, aprender a utilizarlo, conocer el lenguaje de programación usado para iOS (primero Objective-C y después Swift) y crearme mis propias Apps: mis propias herramientas. Es algo así como la personalización perfecta para tu Smartphone.

Pero mi solución no es ni sostenible ni adecuada para la mayoría. No es sostenible, porque cuando creas tu propia App, tienes que mantenerla. A cada nueva versión del sistema operativo, tienes que readaptarla para asegurarte que todo funciona bien. Un trabajo que para una empresa de desarrollo de software es habitual, pero para un particular puede convertirse en una pesadilla. Y no es adecuada para la mayoría, porque yo jugaba con ventaja: ya sabía programar, había trasteado con el kit de desarrollo y además lo considero un hobby. Para quien disfruta de otras aficiones, esta solución es cualquier cosa menos atractiva.

Y aquí es donde Apple ha dado un paso adelante y nos ha traído una herramienta que nos permite dar ese paso más allá de crearnos nuestras propias Apps, sin necesidad de hipotecar nuestro tiempo libre. En este libro, Rafa, José y Frank van a encargarse de ayudarnos a usar Shortcuts, una aplicación que a pesar de su aparente sencillez, esconde un potencial prácticamente ilimitado.

Este libro supone un balance que pocas veces se encuentra: los autores han sabido encontrar, definir y estructurar las ideas clave que definen un programa y su integración completa con Shortcuts, con lo que al terminar el libro puedes utilizar la App de forma natural, teniendo una base sólida.

Y es que estructurando nuestra cabeza adecuadamente y aprendiendo unas nociones básicas de cosas como Markdown, Regex o Scripting, no solo vamos a poder aprovechar nuestro iPhone, iPad y Apps al máximo, sino que de regalo vamos a dejar nuestra mente preparada para aprender a desarrollar en cualquier plataforma.

Bienvenido al “Do it yourself” de las Apps para iOS.

Oliver Nabani es Ingeniero Industrial y creador de contenidos. Es el autor del canal [Oliver Nabani en YouTube](#), dedicado a la tecnología. Suyos son los podcasts [“Todas mis movidas”](#) y

“Mis movidas”. Además es conferenciante sobre cualquier tema relacionado con la tecnología o el diseño aplicado a ella.

Los autores

Frank Liñán García

Nacido en Badalona en 1974 vivió y estudió en la misma ciudad, cursando estudios universitarios en Barcelona.

Frank es Ingeniero Técnico de Telecomunicaciones por la Universidad Politécnica de Catalunya.

Es padre y aprendiz de desarrollador los últimos 25 años, lidiando con tecnología ASP.NET, VB.NET y SQL Server entre otras, obteniendo algunos MOC de Microsoft. Desde siempre ha compaginado su trabajo habitual con proyectos para terceros mediante desarrollo de aplicaciones.

A mediados de los 2000 también hizo incursión en el mundo de la docencia, dando clases para empresas de Excel avanzado y ofimática.

Actualmente trabaja como ingeniero en el sector eléctrico, y en sus ratos libres relata sus experiencias en su podcast [Bateria2x100](#).

Rafael Antonio Roa Hernández

Nacido en Granada (España) en 1980 vivió y estudió en Motril. A los 18 años se trasladó a Granada donde llevó a cabo sus estudios universitarios.

Rafael es Arquitecto Técnico (2009) e Ingeniero de Edificación (2014) por la Universidad de Granada.

Es experto en diseño computacional aplicado a la arquitectura en las plataformas Bentley Genetative Components y Autodesk Dynamo. Ha hecho montajes de arquitectura experimental usando Arduino y Processing, expuestos en el CITA de Copenhague (Dinamarca) y el Rensselaer Polytechnic Institute en Troy, New York (Estados Unidos).

Rafael es cofundador de System Arquitectura (Málaga). Un estudio de arquitectura e ingeniería avanzada donde la simulación y la programación aplicada y la investigación en nuevas tecnologías son la base de los diseños. Con System Arquitectura ha ganado numerosos concursos de arquitectura y la nominación a uno de los premios más prestigiosos del mundo, los premios de arquitectura contemporánea Mies van der Rohe. En System, además de labores de administración desarrolla trabajos de BIM Specialist y Quantity Surveyor.

Además es cofundador de Cortebox. Un servicio de corte láser online sencillo especializado en corte para arquitectura e ingeniería, que pone la fabricación digital al alcance de cualquier persona que tenga conocimientos de CAD o Illustrator.

Secretario/Tesorero de la Asociación por la Arquitectura Avanzada ha organizado para esta tres ediciones de YouFAB, unos cursos de diseño computacional de 4/6 meses de duración para jóvenes de la provincia de Málaga financiados por la Obra Social La Caixa en colaboración con la Diputación de Málaga.

Apasionado de la tecnología desde pequeño comenzó programando en Basic sobre un Spectrum 48k en 1990. Adquirió habilidades en programación html, php, mysql y una larga lista hasta llegar a Objective-C y Swift. Actualmente tiene dos aplicaciones a la venta en la App Store, ambas relacionadas con la arquitectura.

Jose Ruiz Gutierrez

Nacido en Granada (España) en 1980.

Es Técnico Superior en Desarrollo de Aplicaciones Informáticas (2002) por el Instituto de Fomento Ocupacional, y Graduado en Comunicación Audiovisual (2010) por la Escuela Superior de Comunicación en Granada.

Empieza su carrera profesional en el ámbito del desarrollo de software en 2002 al incorporarse al equipo de programación del departamento de Operaciones de Sistemas en HEFAGRA.

En 2004 su carrera da un giro de 180 grados cuando decide adentrarse en el mundo de la comunicación audiovisual para cumplir un sueño: el cine.

Durante los años de formación en la Escuela de Comunicación participa en la producción de numerosos cortometrajes como ayudante de dirección, entre los más relevantes:

- “Inteligente” (2009): Mejor cortometraje en el I Festival Manuel Gutiérrez Aragón en Torrelavega (Cantabria).
- “Veinticuatro” (2010): Mejor cortometraje en el festival CinemaVIP en su edición de 2010 en Marbella (Málaga).

En 2010, participa como ponente en el I Congreso Nacional de Cine Español de Málaga, donde expone la comunicación “Crowdfunding y Creative Commons: nuevos modelos de financiación y propiedad intelectual para la producción y distribución de proyectos audiovisuales”.

Desde 2015 ejerce como Director de proyectos en Miscelánea Audiovisual, empresa de la que es fundador y en la que ha conseguido aunar sus dos facetas profesionales, la informática y el audiovisual, desarrollando proyectos, entre otros, para la Universidad de Granada.

1. Introducción

Hasta hace poco, no eran muchas las opciones de automatización que existían en los dispositivos iOS, el cual Apple ha concebido desde sus primeros días como uno de los sistemas operativos más cerrados en comparación con sus competidores en el mercado, en beneficio de la seguridad y la estabilidad.

Sin embargo, desde la aparición de Workflow el panorama de la automatización en iOS ha mejorado con cada actualización hasta su adquisición por parte de Apple para integrarla en el sistema operativo, momento en el que nace, Shortcuts. Por esto, a partir de ahora Shortcuts puede llegar a donde su antecesor no podía. Ahora, la integración con iOS es cada vez mas profunda y se espera que se vaya ampliando conforme se sucedan las actualizaciones.

Descubriendo Shortcuts para iOS pretende mostrar todas las posibilidades de Shortcuts a aquellas personas que no tienen conocimientos de programación pero desean aprender y sacarle el máximo partido. Pero también se ha pensado en aquellos que ya tienen conocimientos adquiridos y quieren afianzarlos y ampliarlos hasta llegar a hacer shortcuts verdaderamente complejos.

La metodología de **Descubriendo Shortcuts para iOS** cubre, no solo conocimientos sobre Shortcuts, sino que se acerca a otras tecnologías afines y complementarias que no son exclusivas de Shortcuts, y que son ampliamente usadas en otros lenguajes, como por ejemplo RegEx.

Por tanto, puedes ver que tienes en tus manos un libro muy completo que abarca todos los aspectos necesarios para llegar al nivel que desees y que a medida que Shortcuts u otras apps afines introduzcan novedades significativas, estas se verán reflejadas en el libro.

Antes de empezar a aprender debes saber que hay dos recursos en castellano muy importantes en el ámbito de los shortcuts, ambos en la plataforma [Telegram](#):

- [Canal con cientos de shortcuts disponibles para descargar](#)
- [Grupo de discusión y aprendizaje](#), aquí encontrarás una importante comunidad de usuarios de Shortcuts que podrán ayudarte en tus primeros pasos.

Lee este libro con ganas pero sin prisas, date tiempo para asimilar conceptos. Intenta identificar tareas sencillas en las que aplicar el uso de Shortcuts, porque eso te dará la satisfacción y el empuje necesario para llegar hasta el final. A menudo, intentar aprenderlo todo rápido y de golpe provoca saturación, por lo que es mejor que disfrutes del proceso de aprender. Vuelve cuantas veces necesites sobre un capítulo concreto, incluso cuando lo hayas leído entero. Este libro es de aprendizaje pero también de referencia.

1.1. Historia de Shortcuts



ICONO DE WORKFLOW

No se puede hablar de Shortcuts sin antes remontarnos algunos años atrás para conocer cuál fue su origen y cómo ha sido el camino hasta llegar a dónde estamos.

Todo empezó con Workflow, que apareció en la App Store a finales de 2014, causando desde el primer momento un importante revuelo que pronto le aupó a los primeros puestos de aplicaciones más vendidas. En aquel momento la aplicación no era gratuita, costaba 2,99€.

Workflow fue desarrollada por cuatro adolescentes gracias a la cual ganaron la MHACKS de 2014 en Michigan. Ari, Conrad, Veeral, y Nick se proclamaron vencedores de entre 604 proyectos, tomando en consideración aspectos como usabilidad, utilidad, dificultad técnica, y diseño. En el siguiente enlace podrás ver un video (los organizadores de la MHACK han autorizado su uso en este libro) de la presentación, donde aparecen los 4 autores (adolescentes en ese momento):

[Video Hackathon 2014](#)



EL EQUIPO DE WORKFLOW

- Ari Weinstein
- Conrad Kramer
- Veeral Patel
- Nick Frey

Ari Weinstein ([AppleSfera - Entrevista a Ari Weinstein](#)) merece una mención especial. Este joven, ya tiene a sus espaldas un largo recorrido de éxitos. En su corta biografía destaca que a la edad de 6 años era capaz de desmontar y montar una computadora, a los 11 ya “hackeó” su iPod Touch, y a los 15 (2009), fue objeto de entrevistas por formar parte de un grupo de desarrolladores que publicaron un software para poder hacer jailbreak a un iPhone y poder así descargar apps no autorizadas al teléfono. Más de 250.000 descendieron a la página donde lo publicaron para descargarla. [Enlace a la página de Wall Street Journal](#)

En 2013 desarrolló una app llamada [DeskConnect](#) que, básicamente funcionaba de forma similar a la tecnología HandsOff que tenemos actualmente entre macOS e iOS, es decir, era una manera de traspasar ficheros entre Mac y iPhone. La app fue destacada en la AppStore y fue un paso más hacia lo que ocurriría en 2014.

En Enero de 2014, como se ha comentado anteriormente, ganó junto con sus compañeros la MHACKS, con 5.000\$ de premio a repartir. Además, Apple, que era sponsor del evento la galardonó también como la mejor app de iOS del concurso y por ello obtuvieron un iPad Mini cada uno de los 4 integrantes. Esa primera versión se desarrolló en 48 horas, y cómo habrás visto en el video, ya no guarda mucho parecido con la versión actual. Ese fue solo el principio de lo que vendría a posteriori.

Tanto Weinstein como Kramer ganaron también el [Thiel FellowShip](#), un concurso para menores de 20 años que tiene una dotación de 100.000 \$ y que se basa en el lema “*Some ideas just can’t wait*” (Algunas ideas simplemente no pueden

esperar), ayudando a los ganadores a aplazar su etapa estudiantil para poner en práctica dicho proyecto. Weinstein dejaría su Philadelphia natal y Kramer, New Jersey, para trasladarse a San Francisco y dar forma a WorkFlow.

A esta aventura también se unió Nick Frey, dejando atrás Iowa. La primera versión de la app se estrenó en la App store en Diciembre de 2014. Durante 4 días fue la app de pago más descargada en la APP Store, un éxito que les valió centenares de miles de descargas (según apuntaba Ari Weinstein en algunas entrevistas).

A partir de ese momento, actualizaciones y mejoras continuas de la aplicación, añadiendo más y más interacciones y dotando de posibilidades exponenciales a los usuarios avanzados.

Hasta ese momento, herramientas como Drafts, eran de las pocas apps que se se adentraban tímidamente en el mundo de la automatización y la experimentación. Sin duda WorkFlow vino a rellenar un nicho de mercado que estaba por explotar pero que estaba muy limitado debido a las restricciones del sistema iOS, siendo una bocanada de aire fresco para los que piensan que iOS debe y puede ser un sistema operativo muy potente en este tipo de tareas.

Este es el principio de lo que sería una app que fue absorbida por Apple en marzo de 2017. La cantidad no se hizo pública, pero el equipo entero pasó a formar parte de la compañía de la manzana que, en el momento de la venta, estaba compuesto por Weinstein, Kramer, Frey, Patel y Ayaka Nonaka (desarrollador iOS). A partir de ese momento la app pasó a ser gratuita, y sucedió lo que más se temía: el ritmo de actualizaciones descendió. Sin embargo, el temor generalizado que hubo al principio pensando que Apple no haría nada con ella a corto plazo se desvaneció rápidamente. Workflow publicó menos actualizaciones pero no se estancó,

es más, incorporó opciones interesantes y mejoró enormemente otras.

Además, el miedo infundado fue aún mayor debido a la primera actualización después de la venta. En ella, las acciones de Telegram, Line, Uber, Pocket y Chrome fueron eliminadas... Las dos últimas reaparecieron en la versión 1.7.4, en mayo de 2017, y como verás a lo largo del libro, existen alternativas para poder interactuar con alguna de ellas.

Actualmente Nick Frey ha dejado Apple (28/03/2018) para emprender nuevas aventuras.

Veeral Patel sigue en la compañía pero en 2019 tiene pensado estudiar Ciencias de la computación en Standford.

Conrad Kramer, Ayaka Nonaka y Air Weinstein continúan en Apple implicados en el desarrollo de Shortcuts. Ya en la WWDC de esta año el propio Weinstein realizó algunas presentaciones. [Apple Developer Videos 2018](#)

En iOS 12, Apple decidió eliminar Workflow definitivamente y publicar Shortcuts/Atajos en su lugar. Esta integración con iOS era una de las más demandadas por los usuarios y aunque, en la primera versión han quedado aún muchas características fuera, se espera que poco a poco Shortcuts vaya adquiriendo más y más funciones.

1.2. Programación visual

La programación informática es el proceso mediante el cual, a través de una serie de reglas escritas en un determinado lenguaje, se crea un programa que una computadora interpreta para llevar a cabo una acción determinada.

El lenguaje que entienden los procesadores de nuestros dispositivos, ya sean ordenadores, móviles, tabletas, televisores o cualquier otro que tenga un chip, se llama *código*

máquina, que es de muy difícil comprensión a los ojos humanos ya que son cadenas de 0 y 1 muy largas, lo que se llama *código binario*.

Es por esto que aparecieron los lenguajes de programación para escribir el código de una manera más o menos amena para los humanos. Este lenguaje por sí mismo no lo “entiende” el procesador. Para convertir el programa escrito en un lenguaje de programación a código máquina se usa un *compilador*.

Mediante el compilador, un programa es legible para los humanos y para las máquinas. Al mismo tiempo, los compiladores, facilitan que un mismo código se pueda ejecutar en distintos procesadores, ya que cada uno usa su propio código máquina.

A día de hoy hay muchos lenguajes de programación, algunos puede incluso que te suenen: C, Java, Python, Javascript, PHP, Swift, Objective-C, Scratch, C#, VB.NET...

Estos son solo algunos de ellos, hay muchos más, algunos centrados en la eficiencia a la hora de escribir, otros en la facilidad de lectura para los humanos, por lo que la elección del código a usar para programar es una cuestión de comodidad para el programador y de adecuación a las necesidades finales.

Es conveniente diferenciar entre **lenguaje de programación**, mediante el cual se pueden dar instrucciones para que ocurra una cosa u otra en función de unos valores y realizar cálculos y un **lenguaje de marcado**, como HTML, Markdown, LaTeX, BBCode o incluso el RTF, cuyo objeto es dar formato a un texto determinado para incluir elementos como negritas, cursivas, listas, tablas, enlaces o imágenes entre otros.

Recientemente, en un esfuerzo por acercar la programación a todos, están apareciendo lenguajes de *programación visual*. Esto no es más que sustituir las palabras por bloques de

acciones, de manera que enlazándolos unos a otros se pueda proporcionar las instrucciones al programa.

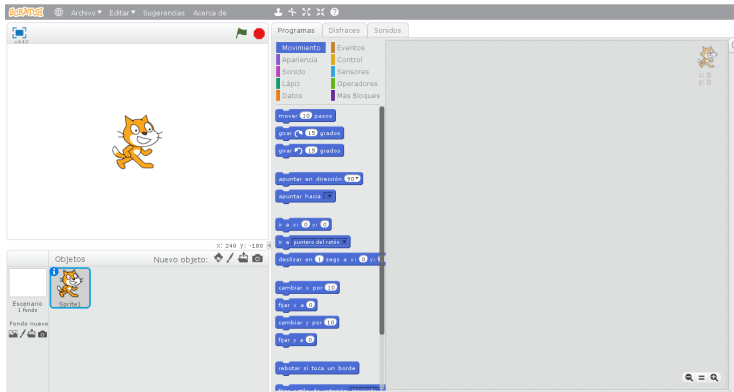
Al ser visual, genera menos rechazo a las personas que se inician a la programación y ciertamente se ve cómo, con su madurez, está empezando a ser una solución muy atractiva incluso para tareas complejas. Prueba de ello es Shortcuts, que mediante programación visual, y tras un pequeño aprendizaje de los conceptos de este libro te va a permitir crear pequeños programas que solucionen tareas repetitivas de tu día a día.

Hay dos lenguajes de programación visual que merecen una mención por cuanto están aportando en este panorama: scratch y blockly.

1.2.1. Scratch

En 2002, apareció [Scratch](#), un lenguaje de programación visual creado en el [MIT Media Lab](#) por el Grupo Lifelong Kindergarten. Su objetivo principal es el de acercar la programación a los más jóvenes, para que les sea sencillo familiarizarse con conceptos de programación que son útiles, independientemente del lenguaje usado posteriormente, ya sea de programación visual o mediante texto.

A lo largo de los años ha ido adquiriendo tal importancia que no solo se está usando para el aprendizaje de los niños, se usa para explicar introducción a la programación a jóvenes y adultos y para la creación de animaciones.



PANTALLA INICIAL DE SCRATCH

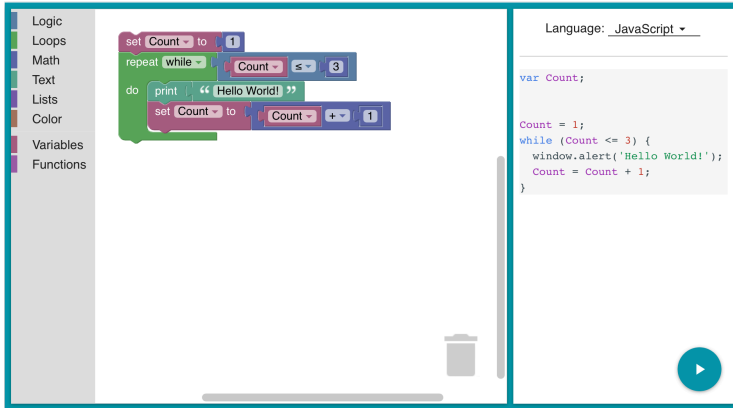
Scratch puede usarse en cualquier entorno de escritorio: Microsoft Windows®, macOS o sistemas UNIX, como Linux o ChromeOS. El instalador puede descargarse [aquí](#). O bien puedes usarlo mediante el [editor online](#). Necesitarás tener Adobe Flash® instalado para su funcionamiento.

1.2.2. Blockly

[Blockly](#) es el otro gran actor en estos momentos. Es una iniciativa de **Google for Education** y a diferencia de Scratch no es un lenguaje de programación visual, sino una interfaz de programación visual, mediante el que puedes arrastrar unos bloques para crear un código que luego puedes convertir al lenguaje que más te guste.

Esto es muy útil en el aprendizaje de un lenguaje nuevo, ya que permite ver las diferencias sintácticas entre los distintos lenguajes que soporta.

Como Blockly no es un lenguaje, sino que es una interfaz, puedes encontrarlo en proyectos muy interesantes como [App Inventor](#), también del MIT que proporciona una herramienta de programación visual basada en Blockly para hacer aplicaciones para Android.



PANTALLA INICIAL DE BLOCKLY

[Blockly Games](#) que permite aprender conceptos de programación mediante pequeños juegos sencillos.

[CODE](#) es otra plataforma en la que aprender a programar mediante juegos como [Flappy Bird](#). Disponen de una amplia oferta de cursos para todas las edades y niveles.

Y muchos otros como [Microsoft MakeCode](#), [OzoBlockly](#) con el que usar blockly para programar el OZO Robot.

2. Conceptos básicos

Tras el lanzamiento de Shortcuts se supo que Apple había decidido traducir su nombre a los distintos idiomas de iOS, por tanto en el mundo hispano su nombre es “Atajos”. A pesar de esto y por facilitar la familiarización con el nombre estándar en este libro se ha convenido usar la nomenclatura inglesa, por lo que nos referiremos a la aplicación Atajos como “Shortcuts”. Un atajo que se ejecute dentro de Shortcuts será un “shortcut” (nótese que la palabra está escrita en minúscula en este caso). Y un atajo para Siri será un “Siri shortcut”.

2.1. Identificar Tareas Repetitivas

“Vale, tengo la aplicación instalada, y ¿Ahora qué?” Esta es la pregunta del millón. Consejo: sigue leyendo y tranquilo, esta pregunta se responderá sola.

Este es el principal freno a Shortcuts, la curva de aprendizaje para poder hacer con ella algo útil es lenta. A menudo verás ejemplos y pensarás: “sí, está muy bien, pero esta no es mi problemática”. Totalmente cierto, es muy difícil saber como usa cada uno de nosotros el iPhone o iPad. Solemos pensar que la manera como realizamos nuestras tareas es cómo lo hacen los demás, y nada más lejos de la realidad.

El identificar “qué puede hacer Shortcuts por ti” es un camino que se va clarificando a medida que usas la aplicación y empiezas a ver posibilidades reales de productividad. Como se ha dicho, y probablemente vuelva a repetirse a lo largo de los capítulos, la posibilidad de integrar Shortcuts con herramientas como Recordatorios de iOS, [DUE](#) e [IFTTT](#) le otorga un carácter que va más allá de una app que abres para ejecutar una tarea. Gracias a estas interacciones, serán estas apps de terceros las que te podrán recordar que hay que ejecutar una serie de acciones, y estas las tendrás en un flujo de trabajo llamado shortcut.

Por ejemplo, imagina que tuvieras que guardar una serie de datos (fecha y hora, un número de horas, cantidad y localización) de una forma más o menos automatizada en una hoja de cálculo. Esto se puede afrontar de dos modos, o bien cada día introduces a mano la dirección, la fecha y hora y los datos en un Excel, lo que implica un par de minutos cada vez que realizas esta tarea; o bien implementas un flujo de trabajo que te facilite esta tarea que realiza 4 o 5 veces al día y te ahorras además de tiempo, el hecho de tener que pensar que acciones debes hacer... Buscar el fichero, abrirlo, situarte en la celda adecuada y rellenar los datos de cada columna, etc... Incluso estarás eliminando posibilidad de equivocarte al añadir dichos datos por un despiste, ya que ahora es la máquina la que trabajará para ti.

Al final del libro se espera que puedas saber identificar qué tareas realizas cada día una o más veces con el teléfono/iPad, y son factibles de llevar a cabo con un simple "tap".

Otro aspecto que se ha podido comprobar es la ayuda que un shortcut puede aportar a una persona invidente. Se han realizado algunos atajos que, de alguna manera, facilitan un poco la tarea a los que no les es tan fácil. Por ejemplo, un shortcut que permite seleccionar una foto y al ejecutarlo describe mediante audio lo que aparece en la fotografía. Este shortcut en particular usa la API de [Microsoft Visio](#) y es realmente interesante, como ya puedes intuir, Shortcuts puede vitaminarse mediante servicios externos. Además, ahora, la posibilidad de lanzar los shortcuts con frases personalizadas facilitará aún más el día a día a esta parte de la sociedad gracias a los Siri Shortcuts.

Otro ejemplo de automatización, un shortcut al que se le pasa un texto que previamente hayas seleccionado, y usando el traductor de Google (no el de Bing, que es el que traduce por defecto en la acción que hay en la app) Shortcuts lo traduce, lo reproduce como audio, y lo deja copiado al portapapeles.

En definitiva, no hay una fórmula mágica para descubrir tareas recurrentes, y tampoco se puede implementar todo lo que se puede imaginar. Recuerda que iOS es un sistema muy poco permisivo para según qué cosas y, aunque casi siempre se puede conseguir mejorar, hay ocasiones en las que simplemente es imposible automatizar lo que se pide.

Acciones del tipo “enviar un mail con un determinado texto cambiando una serie de valores”, “buscar palabras en un pdf y realizar acciones como enviarlo”, o si quieres, cosas más espectaculares cómo “conectar la alarma de casa y encender o apagar luces dependiendo de la hora del día en que lo ejecutes”, son potencialmente convertibles en shortcuts.

Pero sobre todo, una cosa que debe quedar clara es que no existe, aún, una manera de que un shortcut se ejecute de forma automática en función de la hora, o cualquier otro evento que puedas pensar, como por ejemplo que te llegue un correo, un mensaje o cualquier otra cosa. Los shortcuts se ejecutan a voluntad del usuario. A lo largo del libro se explicarán algunas alternativas para mitigar la carencia de esta característica.

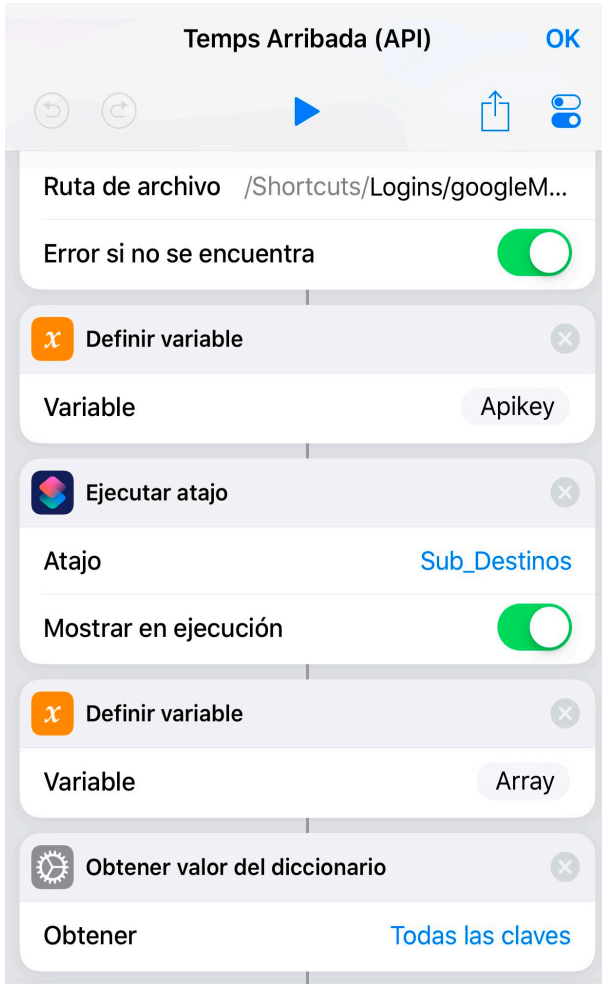
A medida que vayas adentrándote en el libro surgirán posibilidades en tu cabeza que se ajustarán a tu rutina diaria.

2.2. shortcuts vs Siri shortcuts.

Con la aparición de Shortcuts en iOS 12, junto con las recomendaciones que Siri es capaz de hacer cuando realizamos algunas tareas, se ha producido un extraño efecto que a menudo confunde al usuario, no es más que la manera como se llaman a los atajos creados por el usuario, versus los que se exponen en el sistema.

Con la aplicación Shortcuts se pueden crear automatizaciones más complejas, usando los diferentes bloques, APIs, etc... Este libro está orientado a este tipo de shortcuts, donde

seguro encontrarás shortcuts para automatizar tu día a día y aprenderás cómo realizar tus propios flujos de trabajo. Este tipo de shortcuts se pueden también definir con una frase para poder lanzarlos desde Siri. En la siguiente imagen podrás ver una captura de un Shortcut:



SHORTCUT DE EJEMPLO

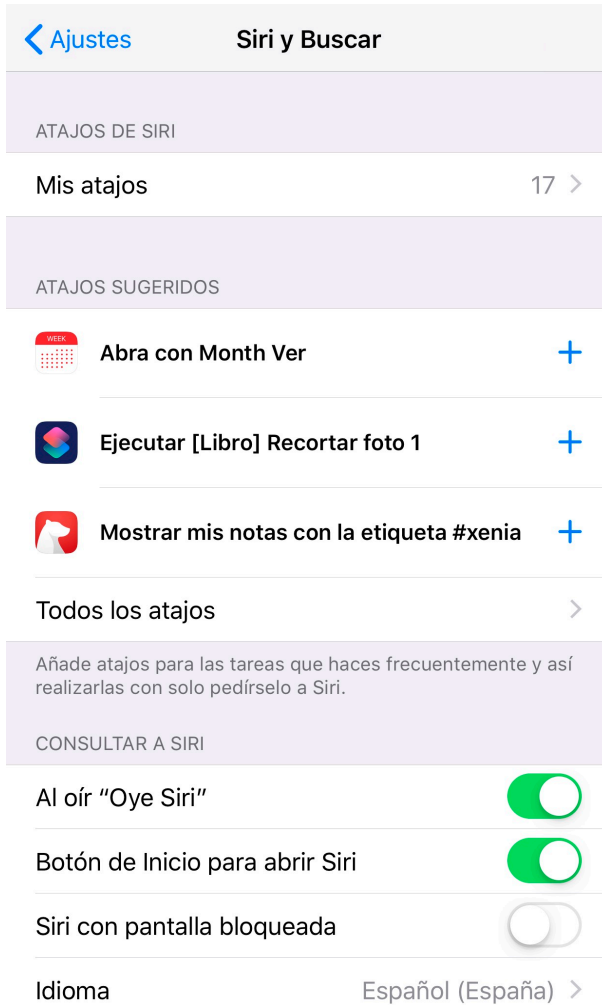
Por otro lado, tenemos los Siri shortcuts (en minúscula) nativos, son definidos en cada app por los desarrolladores y permiten ir a una pantalla concreta de dicha aplicación o incluso realizar tareas en segundo plano, sin necesidad de abrir la app. Un ejemplo de esto último es la app [PCalc](#), donde se pueden realizar pequeñas operaciones o cambios de medida sin la necesidad de abrir la aplicación.

Entre estos dos tipos de atajos está **Siri**, que recomienda “shortcuts” (de ahora en adelante Siri shortcuts) para que sean usados mediante una frase, o añadirlos a tus propios shortcuts como bloques no editables. Es decir, Siri sugiere acciones “simples” que previamente el desarrollador se ha encargado de definir e implementar, mientras que serás tu, amado lector, el que se encargue de desarrollar los shortcuts más complejos.

Estos Siri shortcuts propuestos por Siri no soportan entradas, salidas o variables cuando son añadidas a nuestros shortcuts a medida. Simplemente son pequeñas acciones que funcionan por si mismas, y el desarrollador “ofrece” estas acciones al sistema, ya sea cuando el sistema detecte que repites una determinada acción o a veces, incluso, desde la misma app están listadas todas las posibles opciones que nos ofrece (p.e. Overcast).

Se dispone, pues, de dos tipos de shortcuts, la pregunta inmediata es, ¿Cómo se pueden añadir estos shortcuts que presenta Siri en shortcuts más complejos? Pues bien, en la app de Shortcuts hay una parte llamada “Sugerencias de Siri” donde podrás inspeccionar los shortcuts que están disponibles actualmente.

Para poder ver qué shortcuts ha propuesto Siri puedes acceder a *Ajustes* -> *Siri & Buscar*. En este apartado se muestran todas las aplicaciones que tienen definidos shortcuts, pudiendo añadirlos a tus rutinas diarias mediante una simple frase o palabra.



AJUSTES DE SIRI SHORTCUTS DENTRO DE LOS AJUSTES DE IOS.

Que Siri permita grabar una frase para relacionarla con un *shortcut* o *Siri shortcut*, ofrece una ventaja clara para abordar lo que todos los usuarios han experimentado alguna que otra

vez: Siri no te entiende. Este nuevo paso que ha dado Apple es sin duda un comienzo, pero está lejos de lo que como usuarios avanzados deseáramos, que no es más que poder definir variables en estas frases. Por ejemplo, imagina que pudieras decir algo como “Oye Siri, Acorta URL del portapapeles”, donde “Acorta URL” fuera el nombre del Shortcut, y “portapapeles” fuera un input de ese atajo, es decir, una variable. Esto abriría un mundo de nuevas posibilidades, pero en el momento de escribir estas líneas, esto no es posible.

No cabe ninguna duda que en este momento Apple está en la cola de la carrera de la innovación de los asistentes de voz y que, aunque sin duda están trabajando en mejorar, no tienen aún nada preparado para el gran público, de ahí que los shortcuts sean una manera de mejorar su funcionamiento. Han cambiado de un sistema de lenguaje natural (mediante inteligencia artificial “sabe” lo que quieres sea cual sea la forma de decirlo), por un sistema de “frases pregrabadas” que lanzan Siri shortcuts o shortcuts para cubrir tus necesidades.

La convergencia de esta solución a modo de parche junto con una Siri más inteligente será lo que realmente aportará al ecosistema de Apple la posición que necesita para competir en el futuro de los asistentes virtuales.

2.3. Interfaz y funcionamiento

La interfaz de Shortcuts está diseñada de forma funcional e intuitiva por lo que te resultará fácil dar los primeros pasos dentro de la aplicación.

2.3.1. Biblioteca y galería

Al ejecutar la aplicación encontrarás dos apartados diferenciados:

- la **Biblioteca**, que funciona como un contenedor donde se guarda la colección de shortcuts.
- la **Galería**, donde Apple ha seleccionado multitud de ejemplos y los ha organizado en diferentes categorías.



GALERÍA

La primera vez que ejecutes la aplicación encontrarás que la biblioteca está vacía. Aquí será donde guardarás los shortcuts que vayas creando tú mismo, junto a los que descargues de la galería. Conforme vaya creciendo tu colección, los verás aquí en forma de cuadrícula de diferentes colores, cada una de estas “cajitas” representa un shortcut.

En esta primera pantalla hay varios elementos que destacar. En la parte superior izquierda verás el botón **Editar**, al pulsar sobre él activarás el **modo edición** de la biblioteca, y al hacerlo verás “vibrar” los shortcuts de tu colección. Teniendo activo el modo edición podrás seleccionar los shortcuts pulsando sobre cada uno de ellos. Al tener seleccionado un shortcut lo podrás duplicar o eliminar pulsando sobre los iconos que aparecen en la esquina superior derecha. Otra forma de invocar el modo edición es mantener pulsado uno de los shortcuts.

Es importante tener en consideración que Shortcuts no dispone de una papelera dónde recoger temporalmente los shortcuts que eliminas, por lo que es aconsejable que lo pienses bien antes de eliminar alguno de ellos porque al hacerlo perderás para siempre todo el trabajo que le hayas dedicado.

Si tu colección de shortcuts es muy extensa puede resultar complicado localizarlos a simple vista. Para facilitar esto se dispone de la herramienta **Buscar**, con la que podrás filtrar los shortcuts que contengan en el título la cadena de texto que insertes en la búsqueda.

Cada rectángulo de color representa un atajo, al que podemos asignarle un título descriptivo breve, un icono y un color. Para crear un shortcut nuevo puedes hacerlo de dos maneras, pulsando sobre el símbolo (+) situado en la parte superior

derecha o pulsando en el rectángulo blanco que encontrarás al final de tu colección de shortcuts.

Por último, para ejecutar un atajo bastará con pulsar sobre el rectángulo correspondiente. Para visualizar las acciones de un atajo habrá que pulsar el símbolo (...) situado en la esquina superior derecha de cada uno de los rectángulos.

La segunda parte de la pantalla de inicio de la aplicación Shortcuts corresponde a la Galería, desde ella tendrás acceso a multitud de shortcuts creados por la comunidad de entusiastas de esta aplicación que la propia Apple ha seleccionado, validado y clasificado en categorías.

Esta galería es especialmente útil cuando empiezas con Shortcuts ya que permite conocer ejemplos de todo lo que es capaz esta aplicación. Explorar los ejemplos de cada una de las categorías y observar cómo utilizan las diferentes acciones para conseguir un propósito determinado. Esta es la mejor forma de inspirarte para comenzar a crear tus propios shortcuts.

2.3.2. Modos de uso

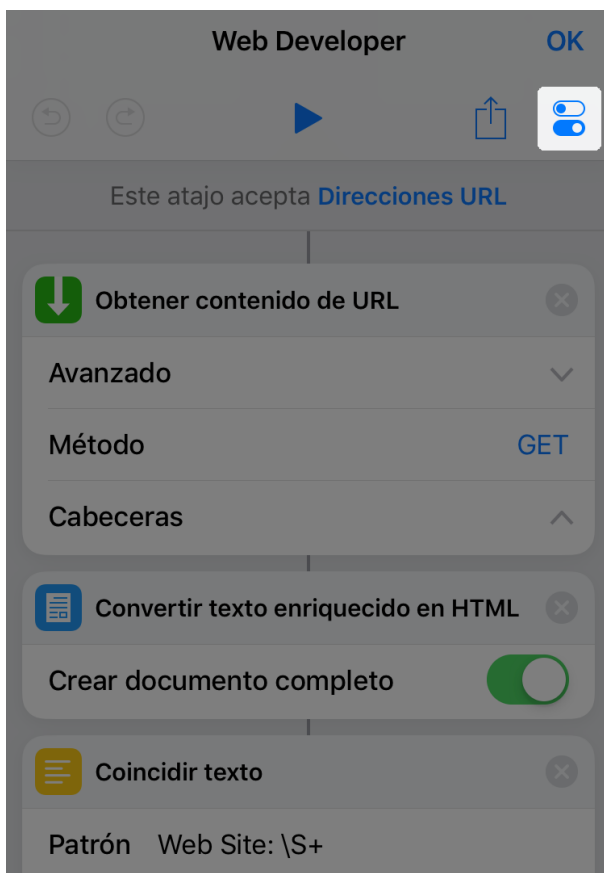
Cómo ya has podido comprobar en la galería, en Shortcuts tienes multitud de opciones para automatizar acciones en tu dispositivo iOS, y cómo no podía ser de otra manera, también existen diferentes circunstancias en las que te resultará útil invocarlos. Por esto la aplicación contempla varias opciones desde las que ejecutar un shortcut que se encuentre en tu biblioteca.

- shortcuts independientes
- widget, siri, icono de escritorio
- shortcuts de extensión
- aplicaciones de terceros

Shortcuts independientes: La opción más sencilla y elemental para lanzar un shortcut es hacerlo como atajo

independiente desde la biblioteca de la aplicación. Esta opción será la que utilices más a menudo mientras construyas, pruebes, y edites tus shortcuts.

Widget, Siri e icono de escritorio: Cuando hayas comprobado que un shortcut funciona como deseas y no contenga errores, puedes ejecutarlo desde las distintas posibilidades que ofrece la interfaz de iOS. Accediendo a los ajustes de un shortcut a través del icono que se encuentra en la parte superior derecha en la vista **editor de shortcuts**, podrás configurar estas opciones.



ACCESO A LOS AJUSTES DEL SHORTCUT

Añadiendo a Siri cualquiera de tus shortcuts podrás ejecutarlo mediante comandos de voz utilizando el asistente de iOS. Esto permite asignar una frase personalizada e invocar los shortcuts a través de “Oye Siri...”.

Cancelar



Añadir a Siri

Ejecutar Web Developer



Puedes decir algo como...

"Web Developer"

Cuando grabes tu frase personalizada, Siri podrá usarla para pedir a Atajos que ejecute este atajo.



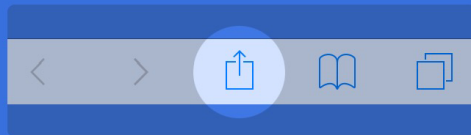
AÑADE UN SHORTCUT A SIRI PARA EJECUTARLO CON TU VOZ

Mostrar en el widget y añadir a pantalla de inicio son opciones muy similares. Ambas crean un acceso directo desde el que ejecutar un shortcut, pero en el primer caso, se accede a él desde el widget de Shortcut (lo que permite lanzarlo aún teniendo el dispositivo bloqueado), la otra opción coloca un icono en la pantalla de inicio que funcionará del mismo modo que cualquier otra aplicación que tengas instalada.

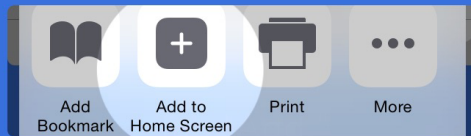


Web Developer

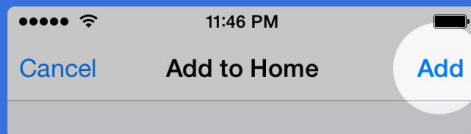
Añadir atajo a la pantalla de inicio



1 Pulsa en la parte inferior de la pantalla.



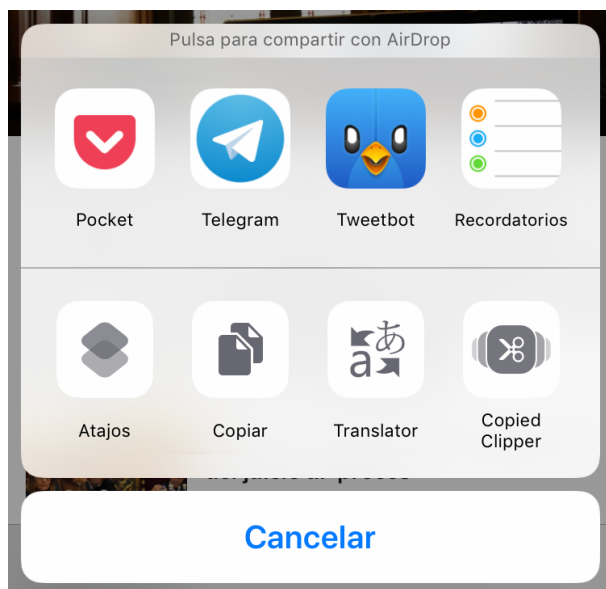
2 Pulsa "Añadir a pantalla de inicio".



3 Pulsa Añadir.

AÑADIR UN ACCESO DIRECTO EN LA PANTALLA DE INICIO

Una de las opciones más interesantes a la hora de ejecutar un shortcut es la que permite hacerlo desde la **extensión de menú**. De esta manera puedes intercambiar información entre aplicaciones. Por ejemplo: obtener información de una determinada web a través de su URL desde Safari, seleccionar una foto del carrito y utilizarla como entrada para un shortcut, etc...



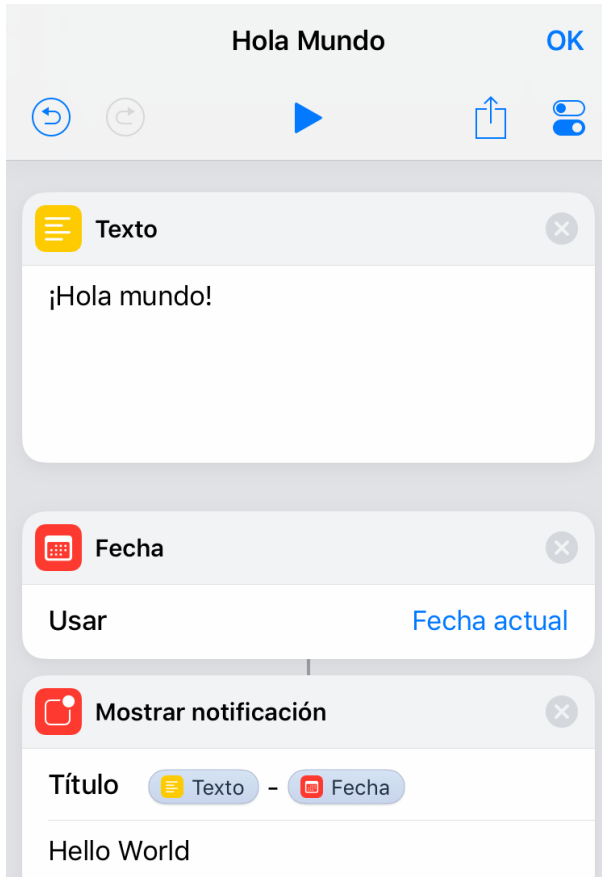
EXTENSIÓN COMPARTIR DE IOS 12

Por último, una referencia a las **aplicaciones de terceros** que son capaces de funcionar como lanzadores de tus shortcuts: Recordatorios, IFTTT, Drafts, DUE, Launcher, y Launch Center Pro. Se abordarán en profundidad en el capítulo "[Cómo ejecutar nuestros shortcuts](#)".

2.4. Como funciona Shortcuts

Un shortcut es un flujo de trabajo, una secuencia de acciones que te permiten automatizar tareas repetitivas con solo pulsar un botón, o incluso de una forma semiautomática, apoyándote en otras apps, como verás más adelante.

Este flujo de trabajo, serie de acciones o receta está compuesto de bloques/acciones. Estos bloques se leen de arriba a abajo, y tienen que tener una lógica entre sí, pero las combinaciones son simplemente infinitas, permitiendo hacer cosas que no serían posibles de otra manera de iOS.



EJEMPLO DE FLUJO DE ACCIONES EN SHORTCUTS

Como se ha explicado en el primer capítulo del libro, si has tenido contacto con algún lenguaje de programación todo será más fácil, pero no es imprescindible para poder realizar shortcuts potentes. La dinámica de la aplicación es simple en cuanto a concepto: Añadir bloques uno tras otro para llegar a un resultado final.

Los actores principales en cualquier shortcut son:

Acciones: Es lo que llamamos bloque. Están definidas y serán la unidad básica para trabajar en cualquier shortcut.



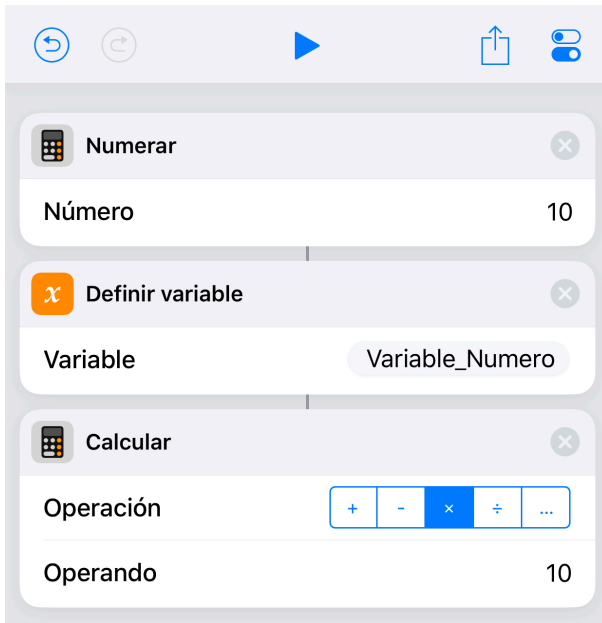
EJEMPLO DE ESTRUCTURA DE UN SHORTCUT

Variables: En el capítulo 4 se hablará en detalle de las variables, pero es el elemento más importante y a veces más confuso de Shortcuts. Una variable no es más que un lugar de memoria donde almacenar un valor, de tal forma que lo puedas usar, modificar, recuperar para tus intereses. Un

shortcut se **lee de arriba a abajo**, no hay saltos (a excepción del bloque “*Ejecuta Shortcut*”, o “*Si*”). Entonces, ¿Cómo puedes recuperar un valor que está varias acciones anteriores al punto actual de ejecución de un shortcut? Para eso están las variables.

En la siguiente captura podrás observar dos acciones enmarcadas, “*Definir Variable*”, y “*Obtener Variable*”. En la primera acción guardamos un valor numérico (siempre se almacena el valor **resultante de la acción anterior**, sea un texto, una foto, un diccionario, etc...).En el ejemplo: 10, con el nombre “*Variable_Numero*”. Es decir, “*Variable_Numero*” vale ahora 10.

Posteriormente se realiza un cálculo, “ 10×10 ”. El resultado el 100, pero en el siguiente bloque NO se utiliza el valor 100 para nada, sino que se recupera el valor de la variable llamada “*Variable_Numero*”, es decir, 10.



SE ALMACENA EL VALOR 10 EN LA VARIABLE Y SE MULTIPLICA POR 10.

Este ejemplo tan poco útil te debe servir para que empieces a entender como funciona Shortcuts. La base, entender que se lee de arriba a abajo. Las variables, sirven para guardar “lo que sea” para poder reutilizarlo más adelante en el shortcut.

Es importante tener en cuenta que cuando se guarda un valor en una variable, ésta se convierte en el valor que almacena. Esto quiere decir que si almacenas el número 10 en la variable “a” y el numero 1 en la variable “b”, podrás hacer una operación tipo a-b cuyo resultado es 9, porque es igual que si hubieras escrito 10-1.

Retomando las acciones, algunas son procesados simples de información, por ejemplo “*Cambiar mayúsculas/minúsculas*”, y otros pueden ser más complejos, como por ejemplo “*Obtener Diccionario de la Entrada*”, que como verás más adelante será de ayuda en numerosas ocasiones, pero implica tener asentados una serie de conocimientos para poder usarlos.

A partir de aquí, se profundizará de lleno en capítulos donde se mezcla lo “fácil” con aspectos más indirectos, como pueden ser las APIs, las expresiones regulares, etc; será gracias a estos conceptos que podrás enfrentarte con éxito a la implementación de shortcuts de un nivel superior.

Un año atrás, un gran hándicap para el uso en general de Workflow, la antecesora de Shortcuts, era el idioma. La aplicación estaba totalmente en inglés, y eso implicó un freno aún mayor para el usuario medio. Eso no ha sido lo más importante, pero sumado a las nociones de programación que se requieren para dar un salto de calidad y poder plantear recetas potentes, se ha provocado un choque entre lo que realmente se puede hacer, y lo difícil que es llevarlo a cabo.

Por otro lado, el 99,9% de la documentación de las APIs está en inglés, por lo tanto, si tu nivel es bajo tendrás más

dificultades. Eso si, con este libro podrás sortear todas estas dificultades o al menos esa es la idea y lo que se intenta es darte multitud de conocimientos y herramientas, junto con ejemplos, para que puedas evolucionar y el idioma no sea un lastre que te impida hacerlo.

La aplicación tiene una característica que la hace genial. Hay pocos menús, botones... pero la potencia es infinita, todo muy Apple.

En los siguientes puntos de este capítulo se van a detallar los diferentes tipos de acciones, el uso de variables y como Shortcuts trabaja con ellas.

3. Acciones

Las acciones son los elementos básicos que conforman un shortcut, estas acciones son pequeños módulos que realizan diferentes tareas sobre los datos que reciben, los transforman, y pasan el resultado a la siguiente acción.

Cada acción realiza un proceso concreto sobre los datos que ha recibido como entrada, la clave para crear un shortcut es entender qué operaciones realiza cada acción, identificar el tipo de resultado que devolverá, y saber organizar las acciones en el orden correcto.

3.1. Concepto de Entrada - Salida

En cualquier programa informático los datos de entrada son aquellos que son procesados. Al finalizar se obtiene los datos de salida que no son más que el resultado derivado de operar a partir de los datos de entrada.

Si se simplifica el modelo de producción de cualquier industria se puede entender el proceso con un ejemplo. Piensa en una fábrica de pan. Esta fábrica recibe en forma de materias primas los ingredientes necesarios para la elaboración del pan (trigo, agua y levadura). Una vez dentro de la fábrica estas materias primas son procesadas. El trigo se tritura hasta convertir el grano en harina, la cual se amasa junto al agua y la levadura, hasta conseguir la masa que finalmente se hornea hasta obtener el pan listo para su consumo.

Si se relaciona el proceso informático con el ejemplo de la elaboración del pan, la fábrica sería el programa y las materias primas los datos estructurados que se van a procesar. La transformación de estos a través de la cadena de producción es comparable a las diferentes funciones que operan en el interior del programa, y el producto elaborado final, el pan, será el resultado obtenido.

Cuando se configura un shortcut como extensión a través del menú compartir, este recibe como entrada la información generada por la aplicación desde la que se le invoca.

En la configuración del shortcut se puede definir el tipo de entrada que acepta. Al usar el menú compartir sólo se mostrarán aquellos shortcuts que admitan el tipo de contenido generado. Ejemplo, ejecutando a la extensión de Shortcuts desde Safari, sólo se mostrarán los shortcuts que admitan una URL como entrada.

< Ajustes

Seleccionar todo

Contactos

Fechas

Direcciones de correo electrónico

Archivos



Imágenes



Productos de iTunes



Ubicaciones

Enlaces de Mapas

Contenido

Archivos PDF



Números de teléfono

Texto enriquecido

Páginas web de Safari

Texto

Direcciones URL



TIPOS DE ENTRADA ACEPTADOS

3.2. Acciones por categoría

El **editor de shortcuts** es el corazón de Shortcuts, desde aquí se muestra un listado con las acciones más importantes. En este listado, cada acción dispone de una ventana de información que muestra una breve descripción, el tipo de datos que es capaz de recibir, y las distintas categorías de resultados que dará como salida.

En un iPad este listado se encuentra en la columna izquierda, mientras que en un iPhone se despliega desde la parte inferior.

En Shortcuts las acciones están clasificadas por categorías en función de su ámbito de actuación.

Al principio aparecen las categorías más usuales con las acciones más comunes, para acceder a todas las opciones deberás pulsar sobre el campo de buscar, entonces aparecerá lo siguiente:

- **Buscar:** es el recuadro con la lupa que hay en la parte superior.
- **Favoritos:** aquí puedes guardar para tener a mano las acciones habituales que usarás en tus shortcuts.
- **Sugerencias de Siri:** muestra acciones frecuentes que hagas cuando usas tu dispositivo, incluso aunque no existan en shortcuts. Es más, aparecerán algunas que no hay otra forma de añadir a shortcuts, como mensajes a grupos de iMessage.
- **Scripts:** aquí están las acciones propias de la programación y que son comunes a cualquier integración con otras aplicaciones.
- **Por temáticas:** A partir de este punto Shortcuts ordena las acciones de varias maneras, por categoría y a continuación por aplicación.

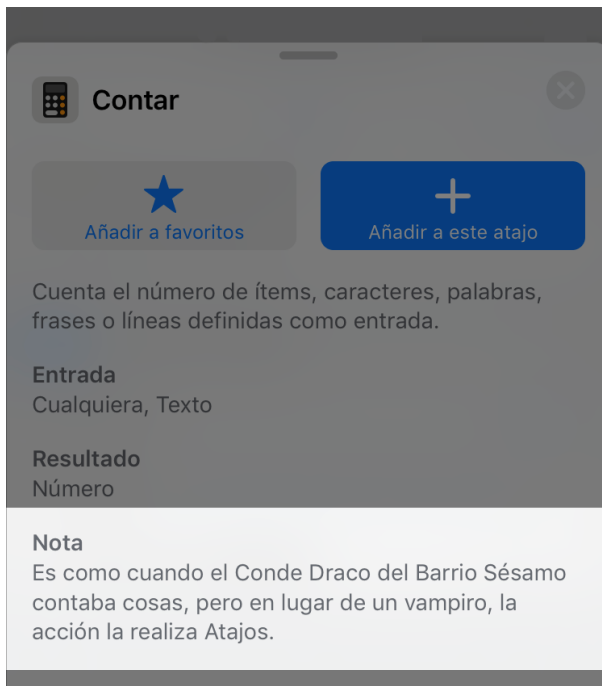
Cuando tengas mucha práctica recordarás el nombre de las acciones que necesitas, y por eso usarás la opción de buscar con más frecuencia. Aún así se van a explicar las diferentes secciones para que te sirva de referencia.

3.2.1. Scripts

Si estás familiarizado con algún lenguaje de programación muchos de estos conceptos te resultarán conocidos ya que están presentes en la mayoría de entornos de desarrollo. Posiblemente se trata del conjunto de acciones más interesantes de las que dispone Shortcuts, ya que permiten crear shortcuts avanzados con funcionalidades propias de aplicaciones nativas.

- **Comentar:** permite insertar un comentario para explicar el funcionamiento de parte del shortcut. Al ejecutarse no hace nada. En el capítulo 3.5 Comentarios puedes encontrar una definición más amplia.
 - Parámetros: comentario
- **Mostrar resultado:** muestra un menú emergente con el texto indicado.
 - Parámetros: texto
 - Salida: muestra un cuadro de diálogo con el texto del parámetro
- **Codificar con Base64:** codifica o descodifica texto o archivos con codificación Base64.
 - Entrada: objeto
 - Parámetros:
 - Modo: codificar o descodificar
 - Saltos de línea: ninguno, cada 64 caracteres, cada 76 caracteres
 - Salida: texto resultado de la codificación y objeto resultado de la descodificación

- **Generar hash:** genera un hash MD5/SHA1 a partir de la entrada.
 - Entrada: archivo
 - Parámetros:
 - Tipo: MD5, SHA1, SHA256, SHA512
 - Salida: texto con el resultado del HASH
- **Contar:** cuenta el número de ítems, caracteres, palabras, frases o líneas definidas como entrada.
 - Entrada: cualquiera
 - Parámetros:
 - Recuento: ítems, caracteres, palabras, frases o líneas
 - Salida: número



ADEMÁS DE GRANDES PROGRAMADORES, EN EL EQUIPO DE SHORTCUTS TAMBIÉN TIENEN UN GRAN SENTIDO DEL HUMOR

- **Obtener nombre:** devuelve el nombre de cada ítem definido como entrada. En función de la entrada, puede ser un nombre de archivo, el título de un sitio web, el título de un evento de calendario, etc.
 - Entrada: cualquiera
 - Salida: texto
- **Obtener tipo:** devuelve el tipo de cada ítem definido como entrada. Por ejemplo, si se define una URL, esta acción devolverá "URL".
 - Entrada: cualquiera
 - Salida: texto
- **Nada:** esta acción no hace nada y no genera ningún resultado. Resulta útil para separar bloques de acciones o para garantizar que ninguna entrada se pase a la siguiente acción.
- **Definir nombre:** ajusta el nombre del ítem definido como entrada.
 - Entrada: cualquiera
 - Parámetros:
 - Nombre
 - No incluir extensión de archivo
 - Salida: cualquiera
- **Ver gráfica de contenido:** Muestra los resultados de la acción anterior en la gráfica de contenido.
 - Entrada: cualquiera
 - Salida: cualquiera
- **Diccionario:** Pasa la lista especificada de pares clave-valor a la siguiente acción como un diccionario. Cuando se realiza

una conversión automática a texto, el diccionario se representa en formato JSON.

- Entrada: diccionario
- Parámetros:
 - Nuevo ítem: texto, número, matriz, diccionario, booleano
- **Obtener diccionario de la entrada:** crea un diccionario a partir del texto definido como entrada. Se pueden utilizar archivos plist basados en XML, JSON (por ejemplo, {"foo": "bar"}) y pares clave-valor (por ejemplo, foo=bar&baz=biz).
 - Entrada: texto, imágenes, diccionarios
 - Salida: diccionarios
- **Obtener valor del diccionario:** obtiene el valor de la clave especificada en el diccionario definido en la acción.
 - Entrada: diccionario
 - Parámetros:
 - Obtener: valor, todas las claves, todos los valores
 - Clave: texto
 - Salida: texto, números, fechas, diccionarios, booleanos
- **Ajustar valor del diccionario:** establece un valor en el diccionario definido en la acción.
 - Entrada: diccionario
 - Parámetros:
 - Clave: texto
 - Valor: texto
 - Salida: diccionario
- **Obtener nivel de batería:** muestra el porcentaje de batería restante como un número entre 0 y 100.
 - Salida: número

- **Obtener dirección IP actual:** devuelve la dirección IP local o externa del dispositivo.
 - Parámetros:
 - Dirección: externa o local
 - Tipo: IPv4 o IPv6
 - Salida: texto
- **Obtener detalles del dispositivo:** obtiene información sobre el dispositivo actual.
 - Parámetros:
 - Obtener: nombre, modelo, versión del sistema, ancho de pantalla, altura de pantalla, volumen actual, y brillo actual
 - Salida: texto, número
- **Obtener detalles de red:** obtiene información sobre las redes a las que está conectado el dispositivo actualmente.
 - Parámetros:
 - Red: Wi-Fi o datos móviles
 - Obtener: nombre de red o BSSID (nombre de identificación único de una red inalámbrica local WLAN)
- **Definir el modo Avión:** define el modo Avión del dispositivo como activado o desactivado.
 - Parámetros:
 - Modo Avión: activado o desactivado
- **Definir Bluetooth:** define la conexión Bluetooth del dispositivo como activada o desactivada.
 - Parámetros:
 - Bluetooth: activado o desactivado
- **Ajustar brillo:** ajusta el brillo del dispositivo. Esta acción define una entrada que devuelve como salida. Si ajustas el brillo como una variable, usa un número entre 0 y 1 (por ejemplo, define 0,5 para un brillo medio).

- **Parámetros:**
 - Brillo: define un porcentaje a través de un control deslizante
- **Definir datos móviles:** define el uso de datos móviles del dispositivo como activado o desactivado.
 - Parámetros:
 - Datos móviles: activado o desactivado
- **Definir “No molestar”:** define el modo “no molestar” del dispositivo como activado o desactivado.
 - Parámetros:
 - No molestar: activado o desactivado
- **Ajustar linterna:** activa o desactiva la linterna situada cerca de la cámara del dispositivo.
 - Parámetros:
 - Linterna: activada, desactivada, o encender/apagar (actúa como un conmutador, si está apagada se enciende, y viceversa)
- **Definir el modo de bajo consumo:** define el modo de bajo consumo del dispositivo como activado o desactivado.
 - Parámetros:
 - Modo de bajo consumo: activado o desactivado
- **Ajustar volumen:** ajusta el volumen del sistema. Esta acción define una entrada que devuelve como salida. Si ajustas el volumen con una variable, usa un número entre 0 y 1 (por ejemplo, define 0,5 para un volumen medio).
 - Parámetros:
 - Volumen: define un porcentaje a través de un control deslizante
- **Definir la red Wi-Fi:** define la red Wi-Fi del dispositivo como activada o desactivada. Esta acción define una entrada que devuelve como salida.

- **Parámetros:**
 - Wi-Fi: activado o desactivado
 - Para conocer mejor el concepto y las acciones referentes a las estructuras de control puedes echarle un vistazo al capítulo 7.5 Estructuras de control.

- **Seleccionar del menú:** presenta un menú y ejecuta diferentes acciones en función del ítem de menú seleccionado.
 - **Parámetros:**
 - Mensaje: instrucción proporcionada cuando se presenta el menú (opcional)
 - Ítems: cada una de las opciones

- **Continuar atajo en la app:** vuelve a Shortcuts y continúa con la acción siguiente.

- **Salir del atajo:** detiene la ejecución del shortcut actual y lo cierra en la pantalla. Después de esta acción, no se ejecuta ninguna más.

- **Si:** Comprueba si algún ítem definido como entrada coincide con la condición especificada y, si es así, ejecuta las acciones que contiene. De lo contrario, se ejecutan las acciones de “Si no”.
 - Entrada: texto
 - **Parámetros:**
 - Entrada: es igual a, contiene, es mayor que, es menor que
 - Valor: cadena de texto con la que se compara la entrada para verificar si se cumple la condición
 - Salida: texto (entrada)

- **Repetir:** repite las acciones incluidas, ejecutándolas el número especificado de veces.
 - Entrada: cualquiera (opcional)

- **Parámetros:**
 - **Repetir:** número de veces que se va a repetir el bucle. Para asignar el número de repeticiones al valor de una variable pulsa el espacio vacío a la izquierda de los símbolos (+) y (-).
- **Salida:** cualquiera

- **Repetir con cada uno:** toma una lista de ítems como entrada y ejecuta las acciones contenidas una vez para cada ítem de la lista.
 - **Entrada:** lista de ítems (opcional)
 - **Salida:** cada ítem pasado a la acción “Terminar repetición”.

- **Esperar:** espera la cantidad de segundos especificada antes de continuar con la siguiente acción.
 - **Parámetros:**
 - **Segundos:** intervalo de tiempo en segundos durante el cual la acción permanece en espera

- **Esperar para regresar:** pausa la ejecución hasta que sales de la app Shortcuts y vuelves a entrar. Resulta interesante de utilizar después de una acción que cambie de app, para poner la ejecución en pausa hasta que regreses a Shortcuts.

- **Seleccionar de la lista:** Presenta un menú con los ítems definidos como entrada.
 - **Entrada:** cualquiera
 - **Parámetros:**
 - **Mensaje:** instrucción que presenta la lista (opcional)
 - **Seleccionar varios:** cuando está activada se pueden seleccionar varios ítems de la lista
 - **Salida:** cualquiera

- **Obtener ítem de la lista:** Devuelve uno o varios ítems de la lista definida como entrada.

- Entrada: cualquiera
- Parámetros:
 - Obtener: Puedes obtener el primer ítem, el último, un ítem aleatorio, el ítem de un índice determinado o ítems de un intervalo de índices
- Salida: cualquiera

- **Crear lista:** Permite especificar una lista de ítems para pasarlos a la siguiente acción.
 - Parámetros:
 - Ítems: puedes añadir tantos ítems como desees. Si especificas una variable, su contenido se incluirá en la lista
 - Salida: cualquiera

- **Calcular:** realiza una operación numérica en la entrada devuelve el resultado.
 - Entrada: cualquier ítem, variable o medición cuyo valor se exprese de forma numérica
 - Parámetros:
 - Operación: suma, resta, multiplicación, división, o operación científica
 - Operación científica: módulo, x^2 , x^3 , x^y , e^x , 10^x , $\ln(x)$, $\log(x)$, \sqrt{x} , $\sqrt[3]{x}$, $x!$, y $\sin(x)$
 - Operando: valor al que aplicar la operación sobre la entrada
 - Salida: número

- **Calcular estadísticas:** calcula las estadísticas de los números introducidos
 - Entrada: cualquier ítem, variable o medición cuyo valor se exprese de forma numérica
 - Parámetros:
 - Operación: media, mínimo, máximo, suma, mediana, modo, intervalo, o desviación estándar
 - Salida: número

- **Convertir medida:** convierte las medidas definidas en la acción a la unidad especificada
 - Entrada: medida
 - Parámetros:
 - Tipo: aceleración, ángulo, área, carga eléctrica, concentración de masa, corriente eléctrica, dispersión, duración, energía, frecuencia, luminancia, longitud, masa, potencia, presión, rendimiento del combustible, resistencia eléctrica, temperatura, tensión eléctrica, velocidad, y volumen
 - Unidad: unidades propias de la magnitud elegida
 - Salida: medida
- **Aplicar formato a número:** aplica un formato a un número (ojo, esto convierte un valor numérico en una cadena de texto). Para asignar el número de decimales al valor de una variable pulsa el espacio vacío a la izquierda de los símbolos (+) y (-).
 - Entrada: booleanos o números
 - Parámetros:
 - Decimales: número de decimales
 - Salida: texto
- **Obtener números de la entrada:** filtra los valores numéricos obtenidos a través de la entrada
 - Entrada: medidas, muestras de entreno, intervalos de tiempo, tamaño de archivos, fechas, tensiones arteriales, boleados, números, contaminantes atmosféricos, muestras médicas, texto
 - Salida: números
- **Medida:** pasa la medida especificada, valor y unidad, a la siguiente acción
 - Entrada: medida

- **Parámetros:**
 - Tipo: aceleración, ángulo, área, carga eléctrica, concentración de masa, corriente eléctrica, dispersión, duración, energía, frecuencia, luminancia, longitud, masa, potencia, presión, rendimiento del combustible, resistencia eléctrica, temperatura, tensión eléctrica, velocidad, y volumen
 - Valor: unidades propias de la magnitud elegida
- Salida: texto

- **Numerar:** pasa un número a la siguiente acción
 - Parámetros:
 - Número: valor numérico (puede ser una variable)
 - Salida: número

- **Número aleatorio:** pasa a la siguiente acción un número aleatorio entre el mínimo y el máximo que se indiquen. El valor mínimo y máximo se incluyen como posibles resultados.
 - Parámetros:
 - Mínimo: valor inferior del rango de posibilidades
 - Máximo: valor superior del rango de posibilidades
 - Salida: número

- **Redondear número:** Redondea los números definidos en la acción.
 - Entrada: número
 - Parámetros:
 - Entero: a la izquierda del decimal, o a la derecha del decimal
 - Modo: normal, redondear siempre hacia arriba, o redondear siempre hacia abajo
 - Decimales: numero de decimales a redondear (puede ser una variable)

- Salida: número
- **Solicitar entrada:** muestra un cuadro de diálogo que indica al usuario que introduzca información
 - Parámetros:
 - Pregunta: mensaje que se muestra al usuario
 - Respuesta por omisión: valor de la respuesta por defecto (opcional)
 - Tipo: texto, número, URL, o fecha
 - Salida: texto, número, URL, fecha
- **Reproducir sonido:** reproduce el archivo de audio definido como entrada o, si no se ha definido ninguno, un sonido de notificación por defecto.
 - Entrada: archivo de audio
- **Mostrar aviso:** muestra un aviso con un título, un mensaje y dos botones. Si el usuario selecciona el botón *Ok*, el shortcut continua. Si selecciona *Cancelar*, se detiene.
 - Parámetros:
 - Título: texto de encabezado
 - Mensaje: cuerpo del mensaje
 - Mostrar el botón Cancelar: activado o desactivado
- **Mostrar notificación:** muestra una notificación
 - Parámetros:
 - Título: texto de encabezado
 - Mensaje: cuerpo del mensaje
 - Reproducir sonido: activado o desactivado
- **Hacer vibrar el dispositivo:** hace vibrar el dispositivo durante un breve periodo de tiempo.
- **Ejecutar script a través de SSH:** ejecuta un script en un ordenador remoto.
 - Parámetros:

- Host: IP del ordenador remoto sobre el que ejecutar el script
 - Puerto: puerto a través del cual se direcciona el servicio SSH, por defecto es el 22
 - Usuario: nombre de usuario en el ordenador remoto
 - Contraseña: contraseña del usuario en el ordenador remoto
 - Script: comando a ejecutar
- **Obtener mis atajos:** almacena los shortcuts incluidos en la biblioteca del dispositivo.
 - Salida: shortcuts
 - **Ejecutar atajo:** llama al shortcut indicado y lo ejecuta, es una manera de reutilizar acciones de forma modular sin repetir los mismos grupos de acciones en diferentes shortcuts
 - Entrada: cualquiera
 - Parámetros:
 - Atajo: shortcut al que se invoca
 - Mostrar ejecución: activado o desactivado
 - Salida: cualquiera
 - **Añadir a variable:** añade el valor de entrada a la variable especificada.
 - Entrada: cualquiera
 - Parámetros:
 - Variable: nombre de la variable (debe estar definida anteriormente)
 - Salida: contenido actualizado de la variable
 - **Obtener variable:** captura el valor de la variable especificada y lo pasa a la siguiente acción.
 - Parámetros:
 - Variable: nombre de la variable (debe estar definida anteriormente)

- Salida: cualquiera
- **Definir variable:** asigna la entrada como valor de la variable especificada
 - Entrada: cualquiera
 - Parámetros:
 - Variable: nombre de la variable
- **Abrir URL X-Callback:** realiza la acción x-callback-url especificada. Los parámetros x-success, x-cancel y x-error se añadirán automáticamente.
 - Entrada: URL
 - Parámetros:
 - Devolución de la llamada personalizada: activado o desactivado
 - URL de x-success personalizada
 - Salida: texto o diccionario. Cuando la app abierta devuelve la llamada a Shortcuts mediante x-success, puede incluir parámetros en la dirección URL. Estos parámetros se definirán como salida en la acción siguiente, como texto en el caso de que solo haya un parámetro o como diccionario si hay varios (usa “Obtener valor del diccionario” para recuperarlo)
- **Codificar con URL:** codifica o descodifica texto definido en la acción para que pueda incluirse en la URL, añadiendo o quitando escapes de porcentaje si es necesario
 - Entrada: texto
 - Parámetros:
 - Modo: codificar o descodificar
 - Salida: texto

3.2.2. Apps

Al momento de publicación de este manual, en esta categoría encontramos únicamente tres acciones; la más importante es

Abrir app que funcionará como un lanzador de aplicaciones, en sus opciones podremos elegir entre cualquiera de las apps que estén instaladas en el dispositivo. Las otras dos nos permiten obtener información de aplicaciones publicadas en la **App Store**.

- **Abrir app**: permite seleccionar cualquier aplicación para que se abra. No tiene opciones de configuración.
- **Obtener detalles de la app de App Store**: a partir de un objeto de App Store, permite seleccionar ciertos detalles a obtener.
 - Entrada: objeto de App Store
 - Parámetros:
 - Artista, precio, código de divisa, precio con formato, fecha de publicación, categoría, descripción, valoración, nº de valoraciones, valoración (esta versión), nº de valoraciones (esta versión), versión, última actualización, notas de la versión, calificación del contenido, versión del SO mínima, es universal, compatible con game center, dispositivos compatibles, idiomas admitidos, direcciones url de capturas de pantalla, direcciones url de capturas de pantalla de iPad, tamaño de la descarga, id de tienda, url de la tienda, ilustración, url de la ilustración, nombre
 - Salida: Texto con la información
- **Buscar en App Store**: Permite buscar en la tienda de aplicaciones una app en concreto.
 - Entrada: ninguna
 - Parámetros
 - Buscar: para introducir el nombre de la aplicación
 - Buscar por: todo, ID del producto, desarrollador, variable mágica

- Resultados: iPhone, iPad, Mac
- País
- N° de resultados
- Salida: objeto de la App Store

3.2.3. Calendario

Cómo podrás adivinar, en esta categoría se encuentran acciones que permiten trabajar con fechas, horas, eventos de calendario y tareas en recordatorios. Especialmente interesantes resultan las acciones que permiten hacer cálculos con fechas y aplicarlos con diferentes formatos. En el capítulo [5.4.1. Trabajando con fechas](#), se aborda en profundidad cómo trabajar con la sintaxis de los diferentes formatos de fecha y hora.

Contactos: otra de las opciones más recurrentes al trabajar con Shortcuts es la de automatizar la búsqueda de contactos, hacer llamadas, o enviar correos y mensajes. Una funcionalidad que resultará interesante en el futuro, ya que aún no está disponible en España, es la posibilidad de enviar y solicitar pagos a través de Apple Pay.

- **Añadir nuevo evento:** Crea un nuevo evento y lo añade al calendario seleccionado.
 - Entrada: ninguno
 - Parámetros:
 - Título: nombre que identifica a este evento
 - Ubicación: (opcional) introduciendo una dirección aquí, la app Calendario podrá determinar el tiempo que tardamos en llegar y establecer alarmas que dependan de esta localización
 - Calendario: elige el calendario al que pertenece el evento
 - Fecha: fecha de inicio, fecha de fin, todo el día

- Aviso: (opcional) determina cuándo establecer una notificación sobre el evento
 - Notas: (opcional) información adicional sobre el evento
- Salida: se crea un evento en el calendario con los datos proporcionados
- **Buscar eventos de calendario cuando:** busca eventos de calendario que cumplan con los criterios seleccionados
 - Entrada: ninguno
 - Parámetros:
 - Fecha de inicio: establece un rango de fecha a partir del cual se inicia la búsqueda
 - Filtros: a través de estos filtros podemos introducir variables que nos ayuden a definir mejor la búsqueda
 - Ordenar por: fecha de inicio o de fin, calendario, duración, título, extensión de archivo, fecha de creación, fecha de última modificación, nombre, aleatorio
 - Límite: variable con la que podemos definir el número máximo de eventos que se obtienen
 - Salida: lista de eventos que coincidan con los parámetros introducidos.
- **Filtrar asistentes al evento:** a partir de una lista de asistentes, devuelve aquellos que cumplen los criterios de búsqueda
 - Entrada: evento de calendario.
 - Parámetros:
 - Filtros: a través de estos filtros podemos introducir variables que nos ayuden a definir mejor la búsqueda
 - Ordenar por: estado, función, nombre, aleatorio
 - Límite: variable con la que definimos la cantidad máxima de asistentes al evento que obtenemos como salida

- Salida: Listado de nombres de los asistentes que cumplan los criterios de filtrado.
- **Obtener detalles de eventos de calendario:** devuelve información sobre los eventos seleccionados
 - Entrada: ninguno
 - Parámetros:
 - Obtener: fecha de inicio y fin, es todo el día, calendario, ubicación, tiene alarmas, duración, organizador, asistentes, URL, título, notas, tamaño del archivo, extensión del archivo, fecha de creación, fecha de última modificación, nombre
 - Salida: listado de eventos que cumplan los parámetros introducidos.
- **Obtener detalles de asistentes al evento:** obtiene información sobre los asistentes seleccionados
 - Entrada: ninguno
 - Parámetros:
 - Obtener: estado, función, soy yo, nombre
 - Salida: texto con los detalles.
- **Obtener próximos eventos:** recupera un listado de eventos ordenados de más próximo a más lejano en el tiempo
 - Entrada: ninguno
 - Parámetros:
 - Calendario
 - Obtener: número de eventos recuperados
 - Día: cualquier día, hoy, mañana, día especificado, variable mágica
 - Salida
- **Eliminar eventos:** elimina los eventos seleccionados
 - Entrada: evento
 - Parámetros:
 - Incluir eventos futuros

- Salida: se eliminan del calendario los eventos de la entrada.
- **Mostrar en el calendario:** muestra la fecha o el evento en la app Calendario
 - Entrada: evento de calendario
 - Parámetros: ninguno
 - Salida: se abrirá la aplicación de calendarios para mostrar el evento.
- **Ajustar fecha:** suma o resta un tiempo determinado a la fecha seleccionada
 - Entrada: fecha
 - Parámetros:
 - Añadir: segundos, minutos, días, semanas, meses o años.
 - Salida: la fecha de entrada con los segundos, minutos, días, semanas, meses o años añadidos.
- **Fecha:** pasa la fecha y hora especificadas a la siguiente acción
 - Entrada: ninguno
 - Parámetros:
 - Usar: fecha actual, fecha especificada
 - Salida: la fecha elegida
- **Aplicar formato a fecha:** establece un determinado formato a la fecha y hora seleccionadas
 - Entrada: objeto fecha.
 - Parámetros:
 - Formato de fecha: ninguno, corta, medio, largo, relativa, RFC 2822, ISO 8601, personalizar, variable mágica
 - Formato de hora: ninguno, corta, medio, largo, relativa, variable mágica
 - Salida: el objeto fecha con el formato de fecha y hora elegidos.

- **Obtener fechas de la entrada:** devuelve cualquier fecha encontrada en la salida de la acción anterior
 - Entrada: texto
 - Parámetros: ninguno
 - Salida: una lista con las fechas que Shortcuts pueda autodetectar en el texto introducido.
- **Obtener tiempo entre fechas:** devuelve la cantidad de tiempo entre la fecha de entrada y la definida en la acción
 - Entrada: objeto fecha.
 - Parámetros:
 - Obtener tiempo desde:
 - Ahora
 - Otros: fecha alternativa
 - Dentro de: la unidad en la que quieres obtener la separación entre las fechas.
 - Salida: segundos, minutos, horas, días,...
- **Añadir nuevo recordatorio:** crea un recordatorio y lo añade a la lista seleccionada
 - Entrada: ninguno
 - Parámetros:
 - Título
 - Lista
 - Avisarme
 - Notas
 - Salida: se crea un recordatorio en la lista especificada con los parámetros dados.
- **Buscar recordatorios:** busca recordatorios que cumplan los criterios definidos
 - Entrada: ninguno
 - Parámetros:
 - Añadir filtro
 - Ordenar por: fecha límite, prioridad, fecha de finalización, lista, título, extensión de archivo,

- fecha de creación, fecha de última modificación, nombre, aleatorio, borrar
 - Orden: en función del filtro seleccionado ofrecerá unas opciones u otras
 - Límite
 - Salida: Lista de recordatorios que cumplen con los criterios de búsqueda.
- **Obtener detalles de recordatorios:** obtiene determinada información de los recordatorios definidos en la acción anterior
 - Entrada: objeto recordatorio
 - Parámetros:
 - Obtener: fecha límite, prioridad, se ha completado, fecha de finalización, lista, tiene alarmas, ubicación del recordatorio, título, notas, tamaño del archivo, extensión del archivo, fecha de creación, fecha de última modificación, nombre, seleccionar variable mágica
 - Salida: texto con el detalle que se haya seleccionado.
- **Obtener próximos recordatorios:** obtiene los próximos recordatorios ordenados de más próximo a más alejado en el tiempo
 - Entrada: ninguno
 - Parámetros:
 - Lista
 - Obtener 1 recordatorio
 - Salida: lista con el número de recordatorios que se haya seleccionado.
- **Eliminar recordatorios:** elimina los recordatorios definidos en la acción anterior de las listas en las que se encuentren
 - Entrada: objeto recordatorio.
 - Parámetros: ninguno.
 - Salida: elimina de la aplicación recordatorios los que se hayan pasado en la entrada.

- **Crear alarma:** crea una alarma en app Reloj
 - Entrada: ninguno
 - Parámetros:
 - Etiqueta
 - Hora
 - Repetir: nunca, lunes, martes, miércoles, jueves, viernes, sábado, domingo, borrar
 - Salida: se creará la alarma con las especificaciones dadas.

- **Iniciar temporizador:** crea una cuenta atrás en la app Reloj
 - Entrada: ninguno.
 - Parámetros:
 - Duración: número en días, horas, minutos o segundos
 - Salida: se creará una cuenta atrás con los datos introducidos.

- **Activar o desactivar alarma:** obtiene el listado de alarmas de la app Reloj y las activa o desactiva según se indique
 - Entrada: alarma
 - Parámetros
 - Alarma: lista de las alarmas actualmente configuradas en tu dispositivo.
 - Activada: estado de dicha alarma.
 - Salida: cambiará el estado de la alarma seleccionada.

3.2.4. Contactos

Otra de las opciones más recurrentes al trabajar con Shortcuts es la de automatizar la búsqueda de contactos, hacer llamadas, o enviar correos y mensajes. Una funcionalidad que resultará interesante en el futuro, ya que aún no está disponible en España, es la posibilidad de enviar y solicitar pagos a través de Apple Pay.

- **Buscar contactos:** busca en la agenda contactos que coincidan con los criterios de búsqueda
 - Entrada: ninguno
 - Parámetros:
 - Añadir filtro
 - Ordenar por: nombre, segundo nombre, apellidos, cumpleaños, tratamiento, sufijo, alias, nombre (fonética), apellidos (fonética), segundo nombre (fonética), empresa, cargo, departamento, extensión de archivo, fecha de creación, fecha de última modificación, nombre, aleatorio, límite
 - Salida: lista de contactos que coincidan con la selección de parámetros.

- **Contactos:** pasa los contactos especificados a la acción siguiente
 - Entrada: ninguno.
 - Parámetros
 - Contactos: permite añadir contactos manualmente
 - Salida: lista de contactos seleccionados.

- **Obtener contactos de la entrada:** obtiene los contactos de salida de la acción anterior
 - Entrada: Objeto.
 - Parámetros: ninguno.
 - Salida: lista de contactos encontrados automáticamente por Shortcuts en el objeto de entrada.

- **Obtener detalles de contactos:** obtiene determinada información de los contactos definidos en la acción anterior
 - Entrada: contacto
 - Parámetros:
 - Obtener: nombre, segundo nombre, apellidos, número de teléfono, dirección de correo

electrónico, dirección, url, cumpleaños, foto de contacto, tratamiento, sufijo, alias, nombre (fonética), apellidos (fonética), segundo nombre (fonética), empresa, cargo, departamento, notas, tamaño del archivo, extensión de archivo, fecha de creación, fecha de última modificación, nombre, seleccionar variable mágica

- Salida: el texto con el parámetro que se haya seleccionado para obtener.
- **Seleccionar contacto:** elige un contacto de tu agenda y lo pasa a la acción siguiente
 - Entrada: ninguno.
 - Parámetros:
 - Varios
 - Salida: el contacto seleccionado al abrirse la agenda del usuario.
- **Dirección de correo electrónico:** pasa las direcciones de correo especificadas a la siguiente acción
 - Entrada: ninguno
 - Parámetros: dirección de correo.
 - Salida: la dirección de correo introducida en los parámetros.
- **Obtener direcciones de correo de la entrada:** devuelve las direcciones de correo encontradas en la salida de la acción anterior
 - Entrada: objeto.
 - Parámetros: ninguno.
 - Salida: listado de direcciones de correo presentes en el objeto de entrada.
- **Seleccionar dirección de correo electrónico:** selecciona una dirección de correo electrónico de tu agenda de contactos y la pasa a la acción siguiente
 - Entrada: ninguno

- **Parámetros:** ninguno
- **Salida:** presentará la lista de contactos del usuario para seleccionar un correo electrónico concreto.

Según países, aunque se muestre esta opción funcionará o no. Para que funcione, no solo debe estar presente **Apple Pay**, sino estar **Apple Pay Cash** disponible en tu país.

- **Solicitar pago:** solicita un pago a las personas seleccionadas, a través de una app para realizar pagos desde tu dispositivo
 - **Entrada:** ninguno
 - **Parámetros:**
 - App
 - Destinatarios
 - Cantidad
 - Abrir en la app
 - Nota
 - **Salida:** solicita un pago al contacto dado y muestra la petición en la app de mensajes según se haya seleccionado “abrir en la app”.
- **Enviar pago:** envía un pago a las personas seleccionadas, a través de una app para realizar pagos desde tu dispositivo
 - **Entrada:** ninguno.
 - **Parámetros:**
 - App
 - Destinatarios
 - Cantidad
 - Abrir en la app
 - Nota
 - **Salida:** envía un pago al contacto dado y muestra el envío en la app de mensajes según se haya seleccionado “abrir en la app”.
- **Llamar:** realiza una llamada telefónica al número definido como entrada

- Entrada: número de teléfono.
 - Parámetros: ninguno.
 - Salida: llamada.
- **Facetime:** realiza una llamada mediante FaceTime al contacto definido en la entrada
 - Entrada: número de teléfono o correo electrónico
 - Parámetros:
 - Tipo de llamada: vídeo, audio
 - Salida: Llamada FaceTime del tipo seleccionado.
- **Obtener números de teléfono de la entrada:** devuelve cualquier número de teléfono encontrado en el resultado de la acción anterior
 - Entrada: objeto.
 - Parámetros: ninguno.
 - Salida: lista de números de teléfono autodetectados por Shortcuts.
- **Número de teléfono:** pasa los números de teléfono especificados a la acción siguiente
 - Entrada: ninguno.
 - Parámetros: número de teléfono.
 - Salida: el número de teléfono introducido.
- **Seleccionar número de teléfono:** selecciona un contacto de tu agenda y pasa su número de teléfono a la acción siguiente
 - Entrada: ninguno.
 - Parámetros: ninguno.
 - Salida: se mostrará la agenda del usuario para seleccionar un número de teléfono.

3.2.5. Documentos

La gestión del almacenamiento de archivos, y obtener y guardar información en ellos es la característica común a este

conjunto de acciones. Un caso especial son las aplicaciones de terceros como Drafts, Bear, o Ulysses que permiten trabajar con sus documentos de forma completa y aprovechar mejor sus funcionalidades.

- **Añadir a archivo:** Añade el texto definido como entrada al final del archivo especificado
 - Entrada: texto
 - Parámetros:
 - Servicio: Dropbox, iCloud Drive, ruta de archivo
 - Modo: añadir, anteponer
 - Crear nueva línea
 - Salida: el archivo de la entrada añadido al archivo especificado en las opciones al principio o al final del mismo.
- **Crear carpeta:** crea una carpeta en el servicio de almacenamiento de archivos que se le indique
 - Entrada: ninguno.
 - Parámetros:
 - Servicio: Dropbox, iCloud Drive, ruta
 - Salida: se crea una carpeta en la ruta y servicio especificado.
- **Eliminar archivos:** elimina los archivos definidos como entrada
 - Entrada: archivos.
 - Parámetros:
 - Confirmar antes de eliminar.
 - Salida: se eliminarán los archivos pasados en la entrada.
- **Obtener archivo:** obtiene archivos de iCloud o Dropbox, desactivando “mostrar selector de documentos” podremos especificar la ruta del archivo que queremos rescatar
 - Entrada: ninguno
 - Parámetros:

- Servicio: Dropbox, iCloud Drive, mostrar selector de documentos
 - Seleccionar varios
- Salida

- **Obtener enlace a archivo:** devuelve un enlace público al archivo definido en la acción anterior
 - Entrada: archivo.
 - Parámetros: ninguno.
 - Salida: se obtiene el enlace público al archivo que se pase en la entrada.

- **Guardar archivo:** guarda archivos en iCloud o Dropbox, para especificar una ruta concreta desactiva “preguntar dónde guardar”
 - Entrada: ninguno
 - Parámetros:
 - Servicio: Dropbox, iCloud Drive
 - Preguntar donde guardar
 - Ruta de destino
 - Sobrescribir si el archivo existe.
 - Salida: Se guardará el archivo en el servicio seleccionado. Si se ha seleccionado “Preguntar dónde guardar” aparecerá una ventana para seleccionar la ruta. Si no, aparecerán las opciones de la ruta de destino y de sobrescribir.

- **Filtrar archivos:** a partir una lista de archivos, devuelve aquellos que coinciden con los criterios de búsqueda
 - Entrada: lista de archivos.
 - Parámetros:
 - Filtros
 - Ordenar por: tamaño del archivo, extensión del archivo, fecha de creación, fecha de última modificación, nombre, aleatorio
 - Límite

- Salida: devolverá la lista de la entrada filtrada y ordenada según los parámetros dados.
- **Aplicar formato a tamaño de archivo:** expresa el tamaño de un archivo en la unidad elegida
 - Entrada: numero procedente del tamaño de un archivo.
 - Parámetros:
 - Formato: Bytes, Kb, Mb, Gb, Tb, Pb, Eb, Zb, Yb o más, seleccionar variable mágica
 - Salida: se hará la conversión entre el formato de entrada y el seleccionado. El cálculo se hará del modo 1000 bytes = 1 KB que es el método que usa apple para un calculo más cercano al usuario del espacio en disco frente a 1024 bytes = 1KB habitual.
- **Obtener detalles de archivos:** obtiene información determinada de los archivos definidos en la anterior acción
 - Entrada: archivo.
 - Parámetros:
 - Obtener: tamaño del archivo, extensión del archivo, fecha de creación, fecha de última modificación, nombre
 - Salida: texto con los datos obtenidos del archivo de la entrada.
- **Abrir en...:** abre la entrada en forma de archivo dentro de la app seleccionada
 - Entrada: cualquier objeto
 - Parámetros:
 - Mostrar menu “Abrir en”.
 - App
 - Salida: muestra la clásica pantalla inferior para abrir un documento en otra aplicación que sea compatible si está seleccionada la opción, si no, se abrirá automáticamente en la app especificada en el parámetro App.

- **Extraer archivo comprimido:** extrae el contenido de un archivo comprimido
 - Entrada: archivo comprimido en los siguientes formatos: zip, tar.gz, tar.bz2, tar.xz, tar, gz, cpio, iso
 - Salida: el contenido del archivo comprimido.
- **Crear archivo comprimido:** crea un archivo comprimido que contiene a los archivos definidos como entrada
 - Entrada: archivos.
 - Parámetros:
 - Nombre del archivo comprimido
 - Formato: zip, tar.gz, tar.bz2, tar.xz, tar, gz, cpio, iso
 - Salida: un archivo comprimido en el formato especificado.
- **Marcar:** edita una imagen o documento PDF con Marcación
 - Entrada: imágenes o pdf.
 - Parámetros: ninguno.
 - Salida: abrirá la pantalla de marcado para que el usuario haga las anotaciones que estime oportunas y devolverá el archivo de la entrada con el marcado que se haya hecho sobre él.
- **Crear PDF:** crea un documento PDF a partir de la entrada al que se le puede añadir opcionalmente un margen de 0,6 cm para facilitar su impresión
 - Entrada: cualquier objeto
 - Parámetros:
 - Incluir margen (útil si la entrada no tiene formato).
 - Incluir: todas las páginas, una página, intervalo de páginas
 - Salida: un archivo pdf con el contenido de la entrada.

- **Imprimir:** imprime el documento de entrada a través de la función AirPrint
 - Entrada: ninguno.
 - Parámetros: ninguno.
 - Salida: abre el cuadro de diálogo de impresión.
- **Abrir en libros:** abre la entrada como un documento PDF en la app Libros
 - Entrada: archivo ePub, iBooks, o PDF
 - Salida: guarda y abre el archivo en la app Libros del dispositivo.
- **Vista rápida:** muestra una previsualización de la entrada
 - Entrada: cualquiera.
 - Salida: abre una pantalla con una previsualización del contenido de la entrada

3.2.6. Salud

Si eres de los que te cuidas y prestas especial atención a lo relacionado con tu estado de salud, tienes varias opciones para automatizar el añadir y obtener muestras en la aplicación *Salud* de iOS. Estás van desde el control de nuestro peso o el índice de masa corporal, hasta el análisis del sueño o los alimentos que consumimos para llevar una dieta equilibrada.

- **Buscar muestras médicas:** devuelve aquellas muestras médicas que coinciden con los criterios de búsqueda
 - Entrada: ninguna.
 - Parámetros:
 - Tipo de muestra
 - Fecha de las muestras
 - Filtros
 - Unidad: contar, porcentaje
 - Agrupar por: ninguno, minuto, hora, día, semana, mes, 3 meses, año
 - Ordenar por: valor, fecha de inicio, fecha de fin, duración, origen, nombre, aleatorio

- Límite
 - Salida: los valores de las muestras que coincidan con los parámetros dados.
- **Obtener detalles de muestra médica:** obtiene una información determinada de las muestras médicas definidas en la acción
 - Entrada: Muestra médica
 - Parámetros:
 - Obtener: tipo, valor, unidad fecha de inicio, fecha de fin, duración, origen, nombre
 - Salida: Valor obtenido de la muestra médica de la entrada.
- **Registrar muestra médica:** añade un punto de datos a la app Salud
 - Entrada: muestra medica.
 - Parámetros:
 - Tipo: Cualquiera de las disponibles en la App Salud.
 - Fecha
 - Salida: Añadirá a la app Salud la muestra que se haya especificado
- **Registrar entreno:** añade un entreno a la app Salud
 - Entrada: ninguna.
 - Parámetros:
 - Tipo: Cualquiera de las disponibles en la App Salud.
 - Fecha
 - Duración
 - Calorías
 - Distanciar (es una errata, se refiere a distancia)
 - Salida: Añadirá el entrenamiento seleccionado junto con sus datos a la App Salud.

3.2.7. Casa

Una de las áreas tecnológicas que más se está desarrollando en los últimos meses es la domótica: aquella que integra la gestión energética, la seguridad, y las comunicaciones en nuestro hogar. Aunque en la versión actual de Shortcuts (2.1) sólo existe una acción *Ejecutar ambiente en Casa*, muy probablemente sea una de las categorías que reciban más atención en próximas actualizaciones.

- **Ejecutar ambiente de Casa:** activa un ambiente de Casa
 - Entrada: ninguna.
 - Parámetros:
 - Casa
 - Ambiente
 - Salida: activa el ambiente seleccionado en la casa que se haya predefinido.

3.2.8. Ubicación

A través de la información que podemos introducir en Shortcuts a cerca de nuestra ubicación, podemos calcular tiempos entre distancias, conseguir una dirección, o información climatológica.

- **Filtrar ubicaciones:** dada una lista de ubicaciones, devuelve aquellas que cumplen los criterios especificados
 - Entrada: ubicaciones
 - Parámetros:
 - Filtros
 - Ordenar por: latitud, longitud, altitud, calle, ciudad, provincia, código postal, país, número de teléfono, nombre, aleatorio
 - Límite
 - Salida: Lista de ubicaciones que cumplan los parámetros de la acción.

- **Obtener ubicación actual:** obtiene la ubicación actual de un dispositivo
 - Entrada: ninguno.
 - Parámetros: ninguno.
 - Salida: usa el aGPS del dispositivo y la triangulación WIFI para ofrecer la dirección de la posición en el momento de la ejecución del shortcut.

- **Obtener detalles de ubicaciones:** obtiene una información determinada de las ubicaciones definidas en la acción anterior
 - Entrada: Ubicaciones
 - Parámetros:
 - Obtener: latitud, longitud, altitud, calle, ciudad, provincia, código postal, país, número de teléfono, nombre
 - Salida: texto con el detalle deseado.

- **Obtener direcciones de la entrada:** vuelve cualquier dirección encontrada en el resultado de la acción anterior
 - Entrada: cualquier objeto con texto
 - Parámetros: ninguno.
 - Salida: autodetecta las direcciones que haya en el texto de entrada y las ofrecerá en una lista.

- **Obtener distancia:** calcula la distancia a la ubicación definida en la acción
 - Entrada: ubicación
 - Parámetros:
 - Desde: ubicación actual, ubicación personalizada
 - Ubicación
 - Tipo de ruta: directa, en coche, a pie
 - Unidad: millas, kilómetros
 - Salida: texto con la distancia entre la ubicación de entrada y la ubicación introducida en la acción.

- **Obtener punto intermedio:** obtiene el punto intermedio entre dos lugares
 - Entrada: ninguna.
 - Parámetros:
 - Primera ubicación
 - Segunda ubicación
 - Salida: ofrece una ubicación intermedia entre ambos lugares
- **Obtener URL de Mapas:** crea una URL para buscar la ubicación, el lugar o el texto definido desde una app de mapas
 - Entrada: dirección o ubicación.
 - Parámetros: ninguno.
 - Salida: url directa a la app Mapas de la ubicación de entrada.
- **Obtener duración del viaje:** calcula el tiempo que se tardará en viajar hasta la ubicación definida en la acción
 - Entrada: ubicación.
 - Parámetros:
 - Desde: ubicación actual, ubicación personalizada
 - Ubicación
 - Modo de transporte: en coche, a pie, transporte público
 - Salida: duración en días, horas, minutos y segundos para el traslado a la ubicación de entrada.
- **Buscar negocios locales:** localiza negocios cercanos a la ubicación actual dentro de un radio definido
 - Entrada: ninguna.
 - Parámetros:
 - Buscar
 - Radio (km)

- Salida: listado de negocios locales en el radio dado que cumplan con el parámetro que se haya introducido en buscar.
- **Mostrar ruta:** visualiza la ruta a la ubicación definida en cualquiera de las apps soportadas (Mapas, Google Maps, Citymapper, Transit o Waze)
 - Entrada: ubicación.
 - Parámetros:
 - App de mapas: mapas, google maps
 - Modo: en coche, a pie, en bici (sólo Google Maps), en transporte público
 - Salida: abre la app deseada de navegación con la ruta preparada según el medio de transporte elegido.
- **Mostrar en Mapas:** abre la app que prefieras (Mapas, Google Maps o Waze) y localiza la ubicación definida
 - Entrada: ubicación.
 - Parámetros:
 - App de mapas: mapas, Google Maps
 - Salida: Muestra la ubicación de la entrada en Mapas o Google Maps.
- **Dirección:** pasa la dirección especificada a la siguiente acción
 - Entrada: ninguna.
 - Parámetros:
 - Línea 1
 - Línea 2
 - Ciudad
 - Provincia
 - Código postal
 - País
 - Salida: objeto de ubicación
- **Obtener el tiempo actual:** obtiene las condiciones meteorológicas actuales en la ubicación especificada

- Entrada: ninguna.
 - Parámetros:
 - En: ubicación actual, ubicación personalizada
 - Ubicación
 - Salida: ofrece un texto con la temperatura actual y el estado del cielo entre paréntesis.
-
- **Obtener detalles de las condiciones meteorológicas:** devuelve determinada información sobre las condiciones meteorológicas definidas en la acción anterior
 - Entrada: condición meteorológica
 - Parámetros:
 - Seleccionar: fecha, ubicación, temperatura, máxima, mínima, sensación térmica, condición, visibilidad, punto de rocío, humedad, presión, cantidad de precipitación, probabilidad de precipitación, velocidad del viento, índice uv, hora de salida del sol, hora de puesta del sol, índice de calidad del aire, categoría de calidad del aire, contaminantes del aire, nombre
 - Salida: ofrece un texto con el valor seleccionado para la ubicación dada
-
- **Obtener previsión del tiempo:** obtiene la previsión del tiempo, a cada hora o diariamente, en la ubicación especificada
 - Entrada: ninguna.
 - Parámetros:
 - En: ubicación actual, ubicación personalizada
 - Ubicación
 - Tipo: cada hora, diariamente
 - Salida: ofrecerá las condiciones de previsión del tiempo para el momento de la ejecución o una lista de 24 elementos con la previsión para las próximas 24 horas en cada elemento de la lista.

3.2.9. Música

Shortcuts puede ser muy recurrido a la hora de gestionar la música que tengamos en el dispositivo o en streaming a través de una suscripción con Apple Music. Desde crear Playlists, añadir canciones, obtener detalles en la iTunes Store de un artista, mostrar en la iTunes Store, hasta interactuar con la reproducción de canciones, y el volumen del dispositivo.

- **Obtener detalles del artista de iTunes:** obtiene información determinada de los artistas de iTunes definidos en la acción
 - Entrada: objeto de iTunes.
 - Parámetros:
 - Obtener: género, tipo, id de tienda, url de tienda, ilustración, nombre
 - Salida: ofrece el texto con el detalle que se haya seleccionado.
- **Obtener detalles del producto de iTunes:** obtiene información determinada de los productos de iTunes definidos en la acción
 - Entrada: objeto de iTunes.
 - Parámetros:
 - Obtener: artista, precio, código de divisa, precio con formato, fecha de publicación, género, duración, transmisible en tiempo real, incluye contenido explícito, descripción, id de tienda, url de la tienda, ilustración, url de la ilustración, nombre
 - Salida: ofrece el texto con el detalle que se haya seleccionado.
- **Buscar en iTunes Store:** busca en iTunes Store los ítems que coinciden con los términos de búsqueda especificados
 - Entrada: ninguna.
 - Parámetros:

- Buscar
 - Categoría: películas, podcasts, música, videoclips, audiolibros, cortometrajes, programas, libros, todo
 - Buscar por: todo, id del producto, artista, compositor, álbum, índice de calificación, canción
 - Resultados: canciones, artistas, álbumes, videoclips
 - País
 - N° de items
 - Salida: elementos de la tienda de iTunes que cumplan los criterios dados.
- **Mostrar en iTunes Store:** muestra los productos de iTunes o las apps de App Store definidas en la entrada en una vista de la tienda
 - Entrada: elemento de iTunes.
 - Parámetros: ninguno.
 - Salida: abrirá la app iTunes para mostrar el elemento de la entrada.
- **Añadir a la lista:** añade los ítems definidos como entrada a la lista de reproducción especificada
 - Entrada: objeto de canción.
 - Parámetros:
 - Lista de reproducción
 - Salida: añadirá el objeto de la entrada a la lista seleccionada en los parámetros.
- **Añadir a “A continuación”:** añadir la música definida como entrada a la cola “a continuación”
 - Entrada: objeto de canción.
 - Parámetros:
 - Reproducir: siguiente, más tarde
 - Salida: añadirá la canción de la entrada a la cola de reproducción actual en el punto que indiques.

- **Buscar música:** busca resultados en la biblioteca de música que cumplan los criterios especificados
 - Entrada: ninguna.
 - Parámetros:
 - Filtros
 - Ordenar por: título, álbum, artista, artista del álbum, género, autor, fecha de inclusión, tipo de contenido, duración, número de reproducciones, número de pista del álbum, número de disco, fecha de publicación, contador de omisión, fecha de última reproducción, valoración, nombre, aleatorio
 - Límite
 - Salida: Ofrecerá una lista con las canciones de Apple Music que cumplan con los parámetros introducidos.
- **Borrar “A continuación”:** elimina el contenido de la cola “A continuación”
 - Entrada: ninguna.
 - Salida: borra la cola de reproducción actual.
- **Crear la lista de reproducción:** crea una nueva lista en la app Música con los ítems definidos como entrada
 - Entrada: elementos de la biblioteca musical
 - Parámetros:
 - Nombre
 - Autor
 - Descripción
 - Salida: Crea una lista de reproducción en la app Música.
- **Obtener canción actual:** devuelve la canción que se está reproduciendo desde la app Música
 - Entrada: la música debe estar sonando usando la app música.

- Salida: objeto música del elemento que se esté reproduciendo.
- **Obtener detalles de música:** obtiene una información determinada de los resultados definidos en la acción
 - Entrada: objeto música.
 - Parámetros:
 - Obtener: título, álbum, artista, artista del album, género, autor, fecha de inclusión, tipo de contenido, duración, número de reproducciones, número de pista del álbum, número de disco, ilustración del álbum, incluye contenido explícito, letra, fecha de publicación, comentarios, es un item en la nube, contador de omisión, fecha de última reproducción, valoración, nombre
 - Salida: texto con la información recabada.
- **Obtener lista de reproducción:** obtiene todas las canciones de la lista de reproducción especificada
 - Entrada: ninguna.
 - Parámetros:
 - Lista de reproducción
 - Salida: obtiene las canciones que forman la lista de reproducción seleccionada.
- **Reproducir música:** comienza a reproducir música desde la app Música
 - Entrada: objeto de música.
 - Parámetros:
 - Aleatorio: desactivada, canciones
 - Repetir: ninguna, una, todas
 - Salida: reproducirá los elementos de la entrada con los parámetros de configuración.
- **Seleccionar música:** abre una ventana de la app Música para seleccionar canciones de la biblioteca musical

- **Parámetros:**
 - Seleccionar varias canciones
- **Salida:** mostrará una pantalla emergente para seleccionar música.
- **Poner música en pausa:** pausa la reproducción de la canción que se está reproduciendo en la app Música
 - **Salida:** pausará la reproducción actual.
- **Ajustar volumen:** indica un nivel de volumen del sistema
 - **Parámetros:**
 - Volumen
 - **Salida:** ajustará el volumen al valor dado mediante la barra deslizada.
- **Saltar hacia atrás:** vuelve a reproducir la canción anterior de la cola de música actual
 - **Parámetros:**
 - Ir a: principio, canción anterior
 - **Salida:** dará un salto hacia atrás al principio de la canción actual o al principio de la anterior.
- **Saltar hacia adelante:** reproduce la siguiente canción de la cola de música actual
 - **Salida:** saltará la reproducción al comienzo de la siguiente canción de la cola de reproducción actual.

3.2.10. Fotos y vídeo

Si bien con el salto de Workflow a Shortcuts la aplicación ha ganado en funcionalidades e integración con el sistema operativo, en el caso del tratamiento de imágenes y vídeo no sólo no ha sido así sino que ha perdido una de las herramientas más útiles que existían en Workflow, la acción *editar imagen*. No obstante se han mantenido otras que facilitan el procesado de imágenes por lotes gracias a la automatización: combinar varias imágenes en una sola,

recortarlas, redimensionarlas, o girarlas son algunas de ellas. En el capítulo 6. Trabajando con archivos multimedia se tratan en profundidad todas estas acciones y sus características.

- **Añadir fotograma a GIF:** añadir una imagen al GIF animado existente definido como entrada, en caso de no existir se crea uno nuevo
 - Entrada: GIF animado
 - Parámetros:
 - Imagen: variable que contenga una o mas imágenes
 - Tiempo de retardo
 - Tamaño automático
 - Salida: añadirá la imagen del parámetro Imagen al GIF animado de la entrada y dará como resultado un nuevo GIF.
- **Obtener fotogramas de la imagen:** divide un GIF animado o una ráfaga de fotos en fotogramas individuales
 - Entrada: GIF animado o ráfaga de fotos.
 - Salida: lista con todas las imágenes que forman el GIF animado de la entrada.
- **Crear GIF:** crea un GIF animado a partir de las imágenes o el vídeo definidos en la acción
 - Entrada: conjunto de imágenes o video.
 - Parámetros:
 - Segundos por foto
 - Bucle infinito
 - Número de bucles
 - Tamaño automático
 - Anchura
 - Altura
 - Salida: crea un GIF animado a partir de las imágenes de la entrada.
- **Convertir GIF en vídeo:** convierte un GIF animado en vídeo

- Entrada: GIF animado.
 - Parámetros:
 - Número de bucles
 - Salida: crea un video con el número de repeticiones del GIF deseado.
- **Grabar audio:** usa el micrófono para grabar audio
 - Entrada: micrófono del dispositivo
 - Parámetros:
 - Calidad de audio: normal, muy alta
 - Iniciar: al pulsar, inmediatamente
 - Finalizar: al pulsar, después de...
 - Duración
 - Salida: graba audio desde el micrófono del dispositivo con los parámetros dados. El resultado es un archivo de audio.
- **Hacer foto:** usa la cámara para hacer fotos
 - Entrada: ninguna.
 - Parámetros:
 - Mostrar previsualización de la cámara
 - Número de fotos
 - Cámara: frontal, trasera
 - Salida: la imagen capturada con la cámara.
- **Grabar video:** usa la cámara para hacer un clip de vídeo
 - Entrada: ninguna.
 - Parámetros:
 - Cámara: frontal, trasera
 - Calidad: baja, media, alta
 - Iniciar: al pulsar, inmediatamente
 - Salida: video resultante de la grabación con la cámara.
- **Combinar imágenes:** combina las imágenes definidas en la acción anterior de forma horizontal, vertical o en cuadrícula
 - Entrada: lista de imágenes

- **Parámetros:**
 - Modo: en paralelo, cuadrícula
 - Dirección: horizontal, vertical
 - Espaciado
- Salida: Combina las imágenes de la entrada según los parámetros de configuración.
- **Recortar imagen:** recorta imágenes en un rectángulo más pequeño
 - Entrada: imagen.
 - Parámetros:
 - Posición: centro, arriba izquierda, arriba derecha, abajo izquierda, abajo derecha, personalizada
 - Anchura
 - Altura
 - Salida: recorta la imagen según los parámetros dados.
- **Invertir imagen:** invierte la dirección de las imágenes horizontal o verticalmente
 - Entrada: imagen.
 - Parámetros:
 - Dirección: horizontal, vertical
 - Salida: da la vuelta a la imagen según la dirección seleccionada.
- **Marcas:** edita una imagen o documento PDF mediante Marcación
 - Entrada: imágenes o pdf.
 - Parámetros: ninguno.
 - Salida: abrirá la pantalla de marcado para que el usuario haga las anotaciones que estime oportunas y devolverá el archivo de la entrada con el marcado que se haya hecho sobre él.
- **Enmascarar imagen:** aplica una máscara a cada imagen definida en la acción, puedes recortar imágenes con forma

de rectángulo redondeado, elipse o icono, o bien aplicar una máscara alfa

- Entrada: imagen
 - Parámetros:
 - Tipo: rectángulo redondeado, elipse, icono, imagen personalizada
 - Salida: la imagen de la entrada recortada por la máscara predefinida con la imagen de máscara personalizada.
-
- **Superponer imagen:** superpone una imagen sobre otra
 - Entrada: imagen.
 - Parámetros:
 - Imagen: variable con una imagen en su interior.
 - Mostrar editor
 - Posición: centro, arriba izquierda, arriba derecha, abajo izquierda, abajo derecha, personalizada
 - Anchura
 - Altura
 - Rotación (Grados)
 - Opacidad
 - Salida: será una nueva imagen en la que la imagen de entrada se colorará en el fondo y la imagen del parámetro se superpondrá según el editor o los parámetros dados.
-
- **Cambiar tamaño de imagen:** ajusta imágenes a una anchura y altura determinadas
 - Entrada: imagen.
 - Parámetros:
 - Anchura
 - Altura
 - Salida: imagen con el nuevo tamaño
-
- **Girar imagen:** gira una imagen hacia la derecha un número determinado de grados

- Entrada: imagen
 - Parámetros:
 - Grados
 - Salida: imagen girada tantos grados como se hayan introducido en los parámetros.
- **Buscar fotos:** busca fotos en la biblioteca que cumplan los criterios especificados
 - Entrada: ninguna.
 - Parámetros:
 - Filtros
 - Ordenar por: anchura, altura, la fecha de la foto, tiempo transcurrido, clase de contenido, duración, frecuencia de fotogramas, extensión de archivo, fecha de creación, fecha de última modificación, nombre, aleatorio
 - Límite
 - Salida: imágenes del carrete que cumpla con los criterios de búsqueda.
- **Eliminar fotos:** elimina las imágenes definidas como entrada de la fototeca del dispositivo (esta acción requiere confirmación antes de realizar el borrado)
 - Entrada: imágenes del carrete.
 - Salida: eliminará las imágenes del carrete de fotografías.
- **Obtener la última importación:** obtiene la importación de fotos más reciente realizada en la app Fotos
 - Salida: lista con las últimas imágenes importadas en el carrete.
- **Obtener las ráfagas más recientes:** obtiene las fotos en ráfaga más recientes del carrete
 - Parámetros:
 - Número de ráfagas

- Salida: lista con las imágenes del número de ráfagas configuradas en los parámetros más recientes.

- **Obtener las Live Photos más recientes:** obtiene las Live Photos más recientes del carrito
 - Parámetros:
 - Número de Live Photo
 - Salida: una lista el número de Live Photos configuradas en el parámetro mas recientes.

- **Obtener las fotos más recientes:** obtiene las fotos más recientes del carrito
 - Parámetros:
 - Número de fotos
 - Salida: una lista con el número de fotos configuradas mas recientes.

- **Obtener las capturas de pantalla más recientes:** obtiene las capturas de pantalla más recientes del carrito
 - Parámetros:
 - Número de capturas de pantalla
 - Salida: una lista con el número de capturas de pantallas más recientes configuradas en los parámetros.

- **Obtener los vídeos más recientes:** obtiene los vídeos más recientes del carrito
 - Parámetros:
 - Número de vídeos
 - Salida: una lista con el número de vídeos más recientes configurada en los parámetros.

- **Guardar en álbum de fotos:** añade las fotos y vídeos definidos en la acción anterior, al álbum de fotos especificado
 - Entrada: imagen o video.
 - Parámetros:

- Album
- Salida: se guardará la imagen o el video de la entrada en el álbum de fotos seleccionado.
- **Seleccionar fotos:** pide que selecciones fotos y/o vídeos de la fototeca
 - Parámetros:
 - Seleccionar varias
 - Salida: una lista con las imágenes seleccionadas.
- **Convertir imagen:** convierte las imágenes definidas en la acción al formato de imagen especificado
 - Entrada: imagen.
 - Parámetros:
 - Formato: jpg, png, tiff, gif, jpeg-2000, bmp, pdf, heif, coincidir con la entrada
 - Calidad
 - Conservar metadatos
 - Salida: imagen convertida según los parámetros de entrada.
- **Filtrar imágenes:** dada una lista de imágenes, esta acción devuelve las imágenes que cumplen los criterios especificados
 - Entrada: imagenes.
 - Parámetros:
 - Filtros
 - Ordenar por: anchura, altura, fecha de la foto, tiempo transcurrido, duración, frecuencia de fotogramas, orientación, tamaño del archivo, extensión de archivo, fecha de creación, fecha de última modificación, nombre, aleatorio
 - Límite
 - Salida: imagenes que cumplen con los requisitos de los filtros ordenadas según los criterios dados.

- **Obtener detalles de imágenes:** obtiene una información determinada de las imágenes definidas en la acción anterior
 - Entrada: imágenes.
 - Parámetros:
 - Obtener: álbum, anchura, altura, fecha de la foto, clase de contenido, tipo de foto, es una captura de pantalla, ubicación, duración, frecuencia de fotogramas, orientación, marca de la cámara, modelo de la cámara, diccionario de metadatos, está oculto, está en favoritos, tamaño del archivo, extensión de archivo, fecha de creación, fecha de última modificación, nombre
 - Salida: texto con el valor del parámetro seleccionado.
- **Obtener imágenes de la entrada:** extrae las imágenes del resultado de salida de la acción anterior, por ejemplo, la ilustración del álbum de una canción o las imágenes de un pagina web
 - Entrada: cualquier objeto.
 - Parámetros: ninguno.
 - Salida: lista con todas las imágenes que haya en el objeto de entrada.
- **Codificar contenido:** vuelve a codificar el contenido definido como entrada con el tamaño especificado, con la opción de convertirlo en audio
 - Entrada: archivo de video.
 - Parámetros:
 - Sólo audio
 - Tamaño
 - Velocidad
 - Metadatos: título, artista, álbum, género, año, ilustración
 - Salida: archivo de video recodificado con los parámetros de configuración.

- **Acortar contenido:** presenta una vista que permite acortar el contenido (audio o vídeo) definido en la acción
 - Entrada: archivo de video.
 - Salida: se presentará una pantalla emergente en el que seleccionar el nuevo inicio y fin del video para hacer el recorte. La salida será un nuevo video.

3.2.11. Compartir

Otra de las características más importantes de Shortcuts es la posibilidad de compartir datos con otras aplicaciones. Algunas de las más importantes nos permiten interactuar con el contenido del portapapeles, enviar mensajes, o correo electrónicos.

- **Publicar en álbum compartido:** comparte una fotografía utilizando los álbumes compartidos de la app Fotos
 - Entrada: imágenes
 - Salida: publicará las imágenes de la entrada en un álbum compartido.

Al trabajar con cuentas de correo electrónico en Shortcuts éste pide configurar la cuenta de correo dentro de la aplicación, no usará las que están ya configuradas en el sistema.

Si usas la cuenta de iCloud tendrás la verificación en dos pasos activada, por tanto, para poder usar esta acción desde Shortcuts deberás crear, dentro de iCloud, mediante la página web, una contraseña de un solo uso para esta autenticación.

Si usas una cuenta de Google el proceso será mucho mas sencillo, pues se te mostrará una ventana emergente para confirmar la identidad y ya estará todo preparado.

- **Enviar mensaje de correo:** presenta una herramienta para redactar correo electrónico
 - Entrada: texto
 - Parámetros:

- Mostrar la hora de redacción
 - De
 - Para
 - Cc
 - Cco
 - Asunto
 - Mensaje
- Salida: Se enviará el correo con los parámetros de configuración.
- **Enviar mensaje:** envía un mensaje en forma de iMessage o SMS
 - Entrada: cualquiera
 - Parámetros:
 - Mostrar el ejecutar: si está seleccionado se abrirá la aplicación mensajes, si no, se enviará en segundo plano.
 - Destinatarios
 - Mensaje
 - Salida: se enviará el mensaje con los parámetros definidos.
- **Crear nota:** comparte la entrada a través de la app Notas
 - Entrada: Texto o imágenes.
 - Salida: se creará una nueva nota con los datos de la entrada.
- **Copiar en el portapapeles:** copia el resultado de la última acción en el portapapeles
 - Entrada: cualquier contenido.
 - Parámetros:
 - Solo local
 - Caduca a
 - Salida: se copia al portapapeles la entrada.
- **Obtener portapapeles:** pasa el contenido del portapapeles a la siguiente acción

- Salida: el contenido del portapapeles.
- **Airdrop:** permite compartir el objeto definido como entrada a través de AirDrop
 - Entrada: cualquier objeto.
 - Salida: ventana para seleccionar al contacto de airdrop.
- **Compartir:** indica que compartas la entrada
 - Entrada: cualquier objeto
 - Salida: ventana emergente para seleccionar cómo compartir la entrada.
- **Compartir con extensiones:** indica que compartas la entrada mediante extensiones de acción y extensiones para compartir proporcionadas por otras apps
 - Entrada: cualquier objeto.
 - Salida: ventana emergente para seleccionar cómo compartir la entrada.
- **Publicar en Facebook:** comparte la entrada en Facebook
 - Entrada: imágenes, URLs, y texto
 - Salida: imágenes, URLs, y texto
- **Publicar en Instagram:** envía la imagen definida como entrada a la aplicación de Instagram y copia el pie de foto en el portapapeles; aunque Instagram ha dejado de admitir el campo subtítulo, la acción copia el contenido en forma de texto al portapapeles para pegarlo fácilmente en Instagram como comentario
 - Entrada: contenido de foto, PDF, código de barras, contacto, imagen, ítem de Giphy, texto enriquecido, ubicación, artista de iTunes, app de App Store, producto de iTunes, atajo, contenido de iTunes, nota de Evernote
 - Parámetros:
 - Pie de foto: (opcional)

- Salida: entrada
- **Publicar en Tumblr:** publica en Tumblr el contenido definido en la acción, es necesario dar acceso a la cuenta de Tumblr
 - Entrada: texto
 - Salida: URL de la nueva publicación
- **Abrir Tweetbot:** abre la página especificada en la app Tweetbot
 - Parámetros:
 - Abrir en: línea de tiempo, menciones, mensajes directos, o favoritos
 - Cuenta: (opcional)
- **Buscar texto:** busca en Twitter, a través de la app Tweetbot, el texto especificado
 - Parámetros:
 - Buscar: texto de búsqueda
 - Cuenta: (opcional)
- **Publicar en Twitter:** publica en Twitter, a través de la app Tweetbot
 - Entrada: texto
 - Parámetros:
 - Cuenta: (opcional)
 - Salida: entrada
- **Ver perfil en Tweetbot:** abre el perfil de usuario de Twitter especificado a través de la app Tweetbot
 - Parámetros:
 - Nombre de usuario: cuenta de perfil que se desea visualizar
- **Publicar en Twitter:** publica la entrada en Twitter
 - Entrada: imágenes, URLs, y texto
 - Salida: entrada

- **Enviar mensaje con WhatsApp:** envía el texto con WhastApp
 - Entrada: texto
 - Parámetros:
 - Destinatario/s: contacto o número de teléfono de la persona destinataria del mensaje
 - Salida: entrada

- **Enviar foto con WhatsApp:** abre en WhatsApp la foto, vídeo, o audio definido en la entrada
 - Entrada: imagen
 - Parámetros:
 - Destinatario/s: contacto o número de teléfono de la persona destinataria del mensaje
 - Salida: entrada

- **Publicar en WordPress:** publica la entrada en un blog de WordPress como nueva publicación
 - Entrada: texto, texto enriquecido, o imágenes
 - Parámetros:
 - Salida: URL de la nueva publicación

3.2.12. Texto

Trabajar con información en forma de texto es una de las características de Shortcuts a las que más veces se recurre, no sólo para realizar búsquedas o unir cadenas, sino también convertir entre distintos formatos, lo que abre un gran abanico de posibilidades para interactuar con esta información.

- **Añadir a nota:** permite añadir texto a la nota que especifiquemos
 - Entrada: texto
 - Parámetros:
 - Nota: nota a la que se añade el texto de entrada
 - Salida: nota actualizada

- **Crear nota:** crea una nota a partir del contenido definido en la entrada
 - Entrada: texto y/o imágenes
 - Parámetros:
 - Mostrar la hoja de redacción: muestra una ventana dónde editar el contenido y seleccionar la nota
 - Salida: nueva nota
- **Detectar idioma con Microsoft:** detecta el idioma del texto proporcionado como entrada
 - Entrada: texto
 - Salida: texto con el idioma que se ha detectado.
- **Dictar texto:** transcribe a texto lo que dices en voz alta y pasa el resultado a la siguiente acción
 - Entrada: micrófono del dispositivo
 - Parámetros:
 - Idioma
 - Parar escucha: tras pausa, tras pausa corta, al pulsar
 - Salida: Texto transcrito a partir de la grabación de voz.
- **Generar código QR:** genera un código QR a partir del texto de entrada
 - Entrada: texto.
 - Parámetros:
 - Corrección de error: baja, medio, cuartil, alta
 - Salida: imagen con el código QR.
- **Obtener nombre de emoji:** obtiene los nombres de los emojis definidos en la acción
 - Entrada: texto con emojis.
 - Salida: lista con los nombres de los emojis reconocidos.

- **Obtener texto de la entrada:** devuelve texto del resultado de la acción anterior, por ejemplo, el nombre de una foto o de una canción, o bien el texto de una web
 - Entrada: cualquier objeto.
 - Salida: texto contenido en el objeto de la entrada.
- **Escanear código de barras/QR:** escanea un código de barras o QR con la cámara y devuelve el texto o la URL encontrados
 - Entrada: Imagen a través de la cámara.
 - Salida: texto con el contenido del código de barras o QR.
- **Mostrar definición:** muestra la definición de la palabra definida en la acción
 - Entrada: texto.
 - Salida: texto con la definición de las palabras de la entrada.
- **Leer texto con voz:** lee el texto introducido en voz alta
 - Entrada: texto.
 - Parámetros:
 - Esperar hasta que acabe
 - Velocidad
 - Tono
 - Idioma
 - Voz
 - Salida: reproducción de audio instantáneo con el texto leído en voz alta. **No es un archivo de audio.**
- **Texto:** pasa el texto especificado a la siguiente acción
 - Parámetros: texto
 - Salida: el texto introducido en los parámetros.
- **Traducir texto con Microsoft:** traduce el texto definido en la acción a otro idioma
 - Entrada: texto.

- Parámetros:
 - De: detectar texto, selección de idioma
 - A: selección de idioma
- Salida: texto traducido.

- **Cambiar mayúsculas/minúsculas:** cambia el texto definido en la acción a MAYÚSCULAS, minúsculas o Mayúscula Al Principio De Cada Palabra
 - Entrada: texto.
 - Parámetros:
 - Mayúsculas/Minúsculas: mayúsculas, minúsculas, mayúscula en todas las palabras, mayúscula al principio de cada palabra, mayúscula al principio de cada frase, intercalar mayúsculas y minúscula
 - Salida: texto con las modificaciones.

- **Combinar texto:** une texto, insertando el separador entre cada unión
 - Entrada: lista de textos
 - Parámetros:
 - Separador: nuevas líneas, espacios, personalizar
 - Salida: todos los textos de la lista unidos en un solo bloque de texto.

- **Corregir ortografía:** corrige automáticamente la ortografía del texto definido en la acción
 - Entrada: texto.
 - Salida: texto con correcciones ortográficas aplicadas.

- **Obtener grupo del texto coincidente:** obtiene el texto que coincide con un grupo de captura determinado o con todos los grupos de captura del resultado de una acción “coincidir texto”
 - Entrada: texto.
 - Parámetros:

- Obtener: grupo del índice, todos los grupos
 - Salida: texto con la coincidencia.
- **Coincidir texto:** busca en el texto definido en la acción coincidencias con una expresión regular
 - Entrada: texto.
 - Parámetros:
 - Patrón (RegEx)
 - Distinguir mayúsculas y minúsculas
 - Salida: texto coincidente.
- **Reemplazar texto:** reemplaza el texto definido en la acción por otro texto
 - Entrada: texto.
 - Parámetros:
 - Buscar texto
 - Reemplazar por
 - Distinguir mayúsculas y minúsculas
 - Expresión regular
 - Salida: texto con la sustitución definida aplicada.
- **Dividir texto:** separa en una lista el texto definido en la acción
 - Entrada: texto.
 - Parámetros:
 - Separador: nuevas líneas, espacios, cada carácter, personaizar
 - Salida: lista

- **Buscar notas:** a partir de una lista de notas, devuelve aquellas que coinciden con los criterios de búsqueda
 - Entrada: lista de notas
 - Parámetros:
 - Ordenar: permite ordenar las notas filtradas por cuerpo del texto, carpeta, fecha de creación, última modificación, nombre, o hacerlo de forma aleatoria
 - Límite: ajusta el número máximo de notas filtradas en la salida
 - Salida: lista de notas filtradas

- **Ver nota:** abre la nota definida como entrada
 - Entrada: nota
 - Salida: nota

- **Convertir RTF en HTML:** convierte el texto enriquecido definido como entrada en texto HTML
 - Entrada: objeto contexto
 - Parámetros:
 - Crear documento completo
 - Salida: texto en código html

- **Convertir RTF en Markdown:** convierte el texto enriquecido definido como entrada en texto Markdown
 - Entrada: texto enriquecido
 - Salida: el texto formateado en Markdown

- **Convertir HTML en RTF:** convierte el texto HTML introducido en texto enriquecido que se puede convertir a su vez en otros formatos
 - Entrada: texto de código html
 - Salida: texto enriquecido

- **Convertir Markdown en RTF:** convierte el texto Markdown introducido en texto enriquecido que se puede convertir a su vez en otros formatos
 - Entrada: texto con etiquetas markdown
 - Salida: texto interpretado en formato enriquecido

3.2.13. Internet

Este es otro de los grupos que cobran más importancia a la hora trabajar con shortcuts que hagan uso de información que esté en la red, ya que se encuentran acciones tan importantes como “Obtener contenido de URL” que es vital para comunicarse con las APIs.

- **Filtrar artículos:** dada una lista de artículos, esta acción devuelve los artículos que cumplen los criterios especificados
 - Entrada: artículos.
 - Parámetros:
 - Filtros
 - Ordenar por: título, autor, fecha de publicación, número de palabras, nombre, aleatorio
 - Límite
 - Salida: lista de artículos que cumplan los criterios de filtrado ordenados según la configuración de los parámetros.
- **Obtener detalles de artículo de Diffbot:** obtiene información determinada de los artículos definidos en la acción
 - Entrada: artículos
 - Parámetros:
 - Obtener: título, autor, fecha de publicación, cuerpo, fragmento, número de palabras, url de la imagen principal, url, nombre
 - Salida: el texto del parámetro obtenido

- **Obtener el artículo de Diffbot de la página web:** obtiene los detalles del artículo, como el texto del cuerpo, el autor, o la fecha de publicación de cada URL definida en la acción
 - Entrada: url.
 - Salida: artículo
- **Expandir URL:** expande y limpia las direcciones URL abreviadas mediante un abreviador de direcciones como TinyURL o Bit.ly
 - Entrada: URL acortada
 - Salida: URL larga
- **Obtener componente de URL:** obtiene una parte concreta de la URL definida en la acción
 - Entrada: URL
 - Parámetros:
 - Componente: esquema, usuario, contraseña, host, puerto, ruta, consulta, fragmento
 - Salida: texto con el componente seleccionado
- **Obtener contenido de URL:** obtiene el contenido de las direcciones URL definidas en la acción. Resulta útil para descargar archivos y contenido web, así como para realizar solicitudes API.
 - Entrada: URL.
 - Parámetros:
 - Avanzado
 - Método: GET, POST, PUT, PATCH, DELETE
 - Cabeceras
 - Salida: texto con los datos obtenidos.
- **Obtener cabeceras de URL:** recupera las cabeceras HTTP de la URL definida como entrada con una solicitud HEAD
 - Entrada: URL
 - Salida: Diccionarios

- **Obtener ítems del canal RSS:** descarga los ítems más recientes de un canal RSS
 - Parámetros:
 - URL
 - Número de ítems
 - Salida: artículos o urls dependiendo del canal RSS
- **Obtener canales RSS de la página:** extrae todas las URL de canales RSS de la URL o pagina web determinada
 - Entrada: URL o texto
 - Salida: Direcciones URL
- **Obtener direcciones URL de la entrada:** devuelve cualquier enlace encontrado en el resultado de la acción anterior
 - Entrada: texto.
 - Salida: lista con todas las URL de la entrada.
- **URL:** pasa la URL especificada a la siguiente acción
 - Parámetros:
 - URL
 - Salida: URL introducida en el parámetro.
- **Buscar en Giphy:** busca GIFs que representen el texto especificado mediante Giphy
 - Parámetros:
 - Buscar
 - Mostrar selector de GIF
 - Seleccionar varios
 - Salida: imágenes GIF animadas
- **Añadir a lista de lectura:** añade a la lista de lectura las direcciones URL definidas en la acción
 - Entrada: URL.
 - Salida: la dirección se añade a la lista de lectura de Safari.

- **Obtener contenido de página web:** extrae el contenido de las páginas web definidas en la acción
 - Entrada: URL.
 - Salida: texto enriquecido con el contenido de la web.
- **Obtener detalles de página web de Safari:** obtiene determinada información de las paginas web de Safari definidas en la acción
 - Entrada: URL.
 - Parámetros:
 - Obtener: contenido de la página, selección de páginas, url de la página, nombre
 - Salida: texto o url con la información obtenida.
- **Abrir direcciones URL:** abre en Safari las direcciones URL definidas en la acción
 - Entrada: URL.
 - Salida: abre la web en una ventana de safari.
- **Ejecutar JavaScript en la página web:** ejecuta JavaScript en una página web de Safari definida como entrada
 - Entrada: URL.
 - Parámetros: código javascript
 - Salida: El resultado del javascript
- **Buscar en internet:** busca en internet el texto definido como entrada
 - Entrada: texto
 - Parámetros:
 - Buscar: Amazon, Bing, DuckDuckGo, eBay, Google, Reddit, Twitter, Yahoo!, YouTube
 - Salida: abre la web correspondiente con el texto de entrada en la búsqueda
- **Mostrar página web:** muestra la web definida dentro de un controlador de visualización de Safari, que permite ver la página web sin cambiar de app

- Entrada: URL
- Parámetros:
 - Acceder al lector de Safari
- Salida: Abre una ventana emergente de Safari y muestra la URL

3.3 Tipos de acciones

Como ya se trató en el capítulo anterior: 3.2 Acciones por categorías, las acciones se pueden agrupar en función del tipo de información con el que trabajan. Pero no es la única posibilidad de diferenciar acciones dentro de Shortcuts.

A la hora de crear un shortcut, **saber utilizar las acciones en orden correcto es fundamental para que este se ejecute correctamente y obtengamos el resultado esperado.** Conforme conozcas en profundidad el funcionamiento de la aplicación te darás cuenta de que la mayor parte de los shortcuts que puedes crear responden siempre a una estructura similar: obtener - transformar - compartir.

Por lo tanto podemos hablar de tres categorías de acciones en función de la fase del proceso en el que nos encontramos, éstas son: acciones de obtención, acciones de transformación, y acciones de compartir.

3.3.1 Acciones de obtención

Las **acciones de obtención** son aquellas que importan los datos al flujo de acciones y serán las primeras en ejecutarse. Realizar correctamente la obtención de datos externos es fundamental para el correcto funcionamiento del shortcut. Dependiendo del tipo de información que deseemos transformar, éstas acciones permiten seleccionar imágenes, obtener texto, e incluso introducir datos directamente mediante un cuadro de diálogo.

3.3.2 Acciones de transformación

A las **acciones de transformación** las podríamos considerar como el núcleo de cada shortcut ya que son estas las que procesan los datos mediante operaciones matemáticas como *calcular*, convertir entre formatos, filtrar contenidos, editar fotografías, combinar texto, o realizar traducciones. En definitiva, son las acciones que dan sentido a un shortcut.

3.3.3 Acciones de compartir

Las **acciones de salida o compartir** son las más comunes dentro del listado de Shortcuts. Mediante estas, la aplicación es capaz de enviar el contenido final a diferentes lugares gracias a la integración con otras aplicaciones y servicios. Publicar en Twitter, crear notas, enviar mensajes, o guardar archivos; son algunas de las acciones más comunes dentro del grupo compartir.

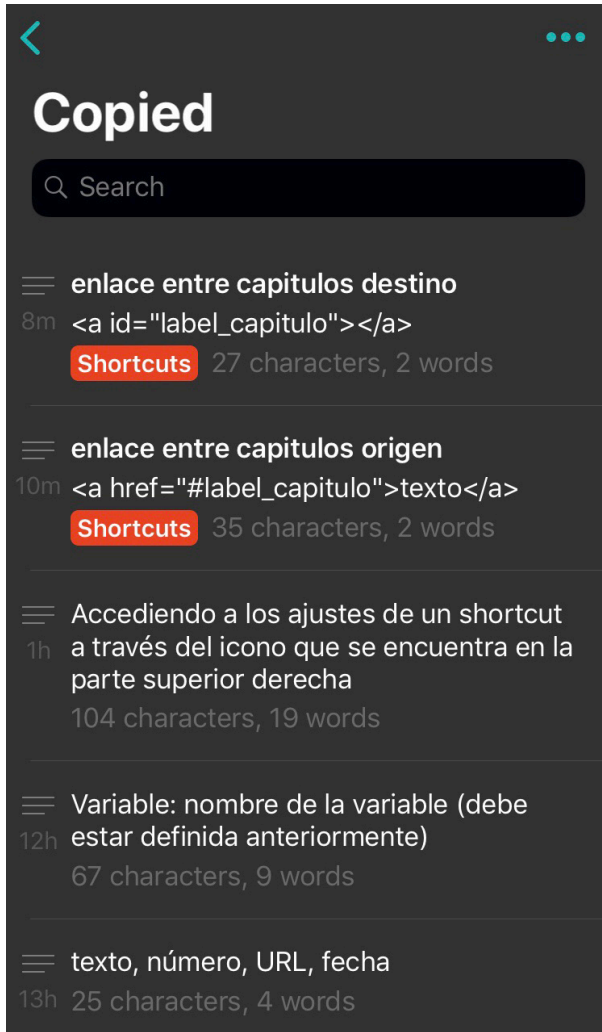
3.4 Relación de Shortcuts con otras Apps: URL-Schemes

Dentro de las posibilidades que nos ofrece Shortcuts para compartir información con otras aplicaciones merecen especial atención las *llamadas a través de URLs*. La perseverancia de Apple por hacer de iOS un sistema estable y seguro tiene algunas contraprestaciones, entre ellas la dificultad de las aplicaciones para comunicarse entre sí. Con la implementación del *sandbox* en iOS, las aplicaciones quedaron “encerradas” y perdieron la posibilidad de realizar operaciones fuera de su ámbito. Afortunadamente los ingenieros de Apple dejaron abierta una posibilidad de interacción entre las aplicaciones a través de **esquemas URL**.

Usando esquemas URL es posible establecer referencias a la información contenida en una aplicación, por ejemplo abrir una fotografía dentro de una aplicación de biblioteca de fotos.

Para habilitar este procedimiento de comunicación entre aplicaciones es necesario que el desarrollador implemente y defina cada una de estas acciones en su aplicación, afortunadamente cada día son más las aplicaciones de la App Store que se ocupan de prestar estas acciones. iOS provee a los desarrolladores y sus apps de métodos para que puedan exponer algunos métodos mediante esta técnica.

URL-Schemes: en un primer peldaño de este tipo de procedimientos se encuentran las *url-schemes*, son aquellas que mediante una url del tipo `nombreapp://` son capaces de lanzar una aplicación y pasarle determinados argumentos cumpliendo una determinada sintaxis. Observa por ti mismo las posibilidades que ofrece con un ejemplo sencillo y útil.



INTERFAZ DE COPIED

En este [enlace](#) encontrarás un listado actualizado de todas URL-Schemes más populares. Otro recurso online interesante es el de x-callback-url.com, donde hay un listado de apps con enlaces a sus ayudas para ver la documentación al respecto.

Las URL-Schemes tienen una sintaxis idéntica a la de las API's en las que podrás profundizar en el capítulo 8.

Por ejemplo, a través de la app [Copied](#) se puede hacer una gestión avanzada del portapapeles del sistema. Lanzándola desde Shortcuts a través de *url-scheme* conseguirás resultados interesantes:

Abrir **Copied** y mostrar el contenido del portapapeles

```
copied://clipboard
```

Abrir **Copied** y mostrar la lista que coincida con “nombreLista”

```
copied://list/[nombreLista]
```

Realizar búsquedas entre el contenido de **Copied** que coincidan con la cadena “terminoBusqueda”

```
copied://search?q=[terminoBusqueda]
```

X-Callback-URLs: un paso más allá van las *x-callback-url*, a las que se podrían considerar “hermanas mayores” de *url-scheme*, ya que además de pasar información entre aplicaciones son capaces de establecer un control sobre el resultado de salida, devolviendo a la aplicación original la verificación correspondiente si el proceso a concluido correctamente o recuperar el valor del resultado final para utilizarlo en la aplicación original, los llamados *action parameters*.

Guardar un elemento en el historial de **Copied** y volver a Shortcuts.

```
copied://x-callback-url/save?  
title=[Titulo]&text=[Texto]&url=[url]&list=[lista]  
&x-success=shortcuts://
```

Esta URL-Scheme recupera el elemento X que se haya puesto en *índice* y lo copia al portapapeles, una vez hecho, como en el parámetro x-success se ha especificado la URL-

Scheme “shortcuts://” si se ha conseguido se volverá a Shortcuts para continuar con el flujo de ejecución sabiendo que el dato que se ha obtenido de Copied estará en el portapapeles.

```
copied://x-callback-url/copy?  
index=[indice]&list=[lista]&x-success=shortcuts://
```

Las diferentes partes de una llamada a mediante x-callbackURL son:

```
[scheme]://[host]/[action]?[x-callback  
parameters]&[action parameters]
```

- **scheme:** Es un string único que identifica a la app. Los esquemas URL son registrados en IOS a través de un fichero llamado info.plist.
- **host:** Las URLs se identifican mediante la cadena “x-callback-url”
- **action:** El método que se expone, depende del desarrollador y debería ser comentado en los Help para que el usuario sepa como utilizarlo.
- **Parámetros x-callback:** Los parámetros de esta parte son opcionales, son pasados mediante argumentos del tipo query en la URL siguiendo el formato “key1=valor1&key2=valor2”. Todos los “valores” deben estar “[URL encoded](#)”, es decir, sustituyendo determinados caracteres por “%{dos números}” para que no haya problemas en la transmisión de la información.
 1. x-source: El nombre(lo más identificable posible) de la aplicación.
 2. x-success: Si la acción pasada devuelve alguna respuesta, esta va en esta parte. Si todo a funcionado correctamente, el resultado va aquí. (en formato URL)
 3. x-error: Igual que el anterior pero en el caso contrario, que haya habido algún error.

4. x-cancel: URL a abrir si la acción pasada se ha cancelado por algún motivo.

Las URL-schemes pueden usarse de dos maneras, en primer lugar deberás añadir una acción “URL” en la que estará la dirección de la URL-Scheme tal y como quieras configurarla y añadiendo los parámetros, que a menudo serán variables o variables mágicas recuperadas del propio flujo del shortcut.

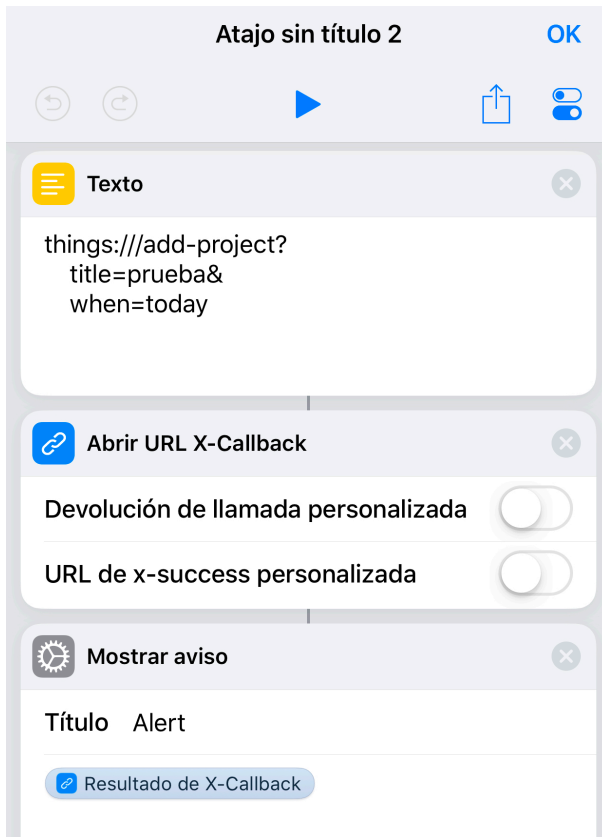
Y en el siguiente paso puedes usar una acción “Obtener contenido de la URL” para que se ejecute la petición de la URL y se ponga en marcha el mecanismo de la URL-Scheme o bien usar la acción “Abrir URL X-Callback” que permite personalizar más como se gestionará la vuelta a Shortcuts tras el éxito o el error al hacer la acción de la URL-Scheme.

La mayor parte de las veces lo que pretenderás será recuperar un dato y se usará el caso de éxito x-success de X-Callback pero no olvides que un error no significa necesariamente que algo es malo. Imagina que en una app del tiempo, mediante una URL-Scheme pudieras hacer una consulta para que te devuelva si en ese día hará sol. Una respuesta de éxito significaría que sí, que va a hacer sol, y una respuesta de error significaría que va a llover.

Es por eso que debes ser tu quien controle como son las respuesta tanto en el éxito como en el error para que las interpretes correctamente y de cara al usuario brindes la información ya adaptada y comprensible de lo que está ocurriendo en tu shortcut.

A continuación vas a ver unos ejemplos del uso de URL-Schemes.

3.4.1 Ejemplo URL-Scheme Things



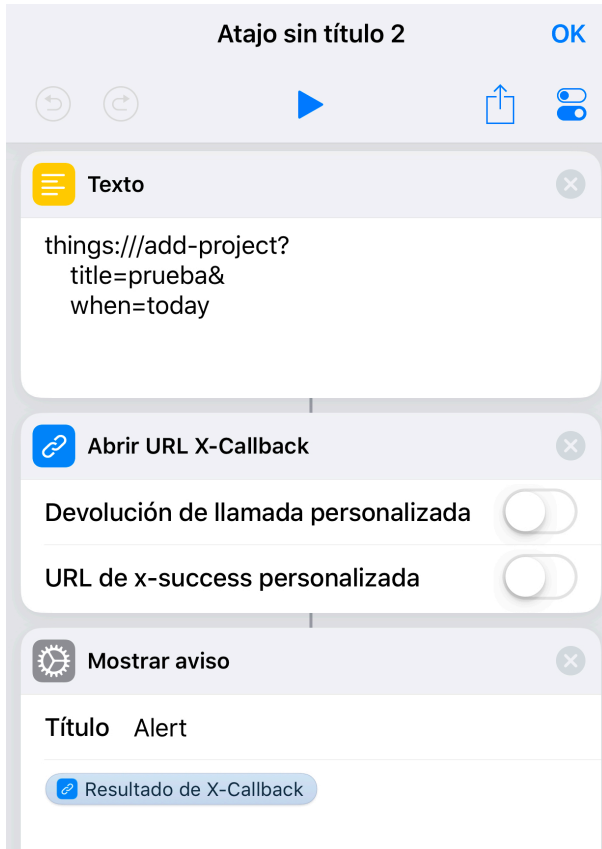
RESPUESTA DE UN X-SUCCESS

Things es un gestor de tareas que sigue la metodología GTD (Getting Thing Done) que viene a ser “Conseguir que se hagan las cosas” o según la traducción del libro de Paul Allen al español “Trabajar con eficacia”.

Observa como la aplicación [Things](#) detalla en su ayuda una cantidad ingente de posibilidades con x-callback-url, y además pone a disposición del usuario el llamado “[link builder](#)”

(creador de enlaces) para poder construir de una forma más amigable estas estructuras. Sin duda, una visita obligada si deseas aprender como funciona esta tecnología.

Siguiendo el ejemplo, en este caso crea el proyecto y devuelve el identificador de este.



RESULTADO DEL SHORTCUT

Las llamadas x-callback-url son una especificación creada por Greg Pierce, desarrollador y presidente de la compañía de

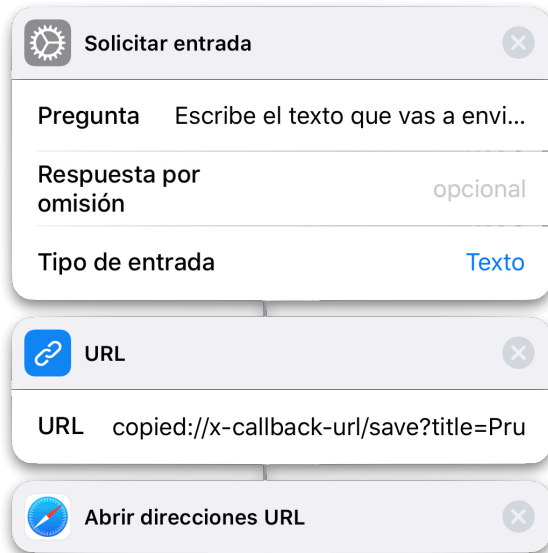
desarrollo [Agile Tortoise](#). Cuando Greg empezó a desarrollar Terminology junto a Marco Arment y Justin Williams, descubrieron que necesitaban implementar un procedimiento que permitiera operar en una aplicación la información requerida desde otra. Esto permitió a los usuarios de Instapaper buscar palabras en Terminology, y estandarizó el modo en que los desarrolladores debían implementar este tipo de integraciones entre aplicaciones.

De no ser por las llamadas x-callback-url, no se habría desarrollado todo el potencial de aplicaciones como Workflow, y seguramente no existiría Shortcuts tal y como lo conocemos hoy.

3.4.2 Ejemplo URL-Scheme Copied

En el siguiente ejemplo se va a ver cómo interactuar con Copied que puedes descargar [aquí](#).

Primero se va a enviar un texto al portapapeles de Copied, para ello usa una acción “Solicitar entrada” para que en la ejecución se te pregunte qué texto quieres enviar a Copied.



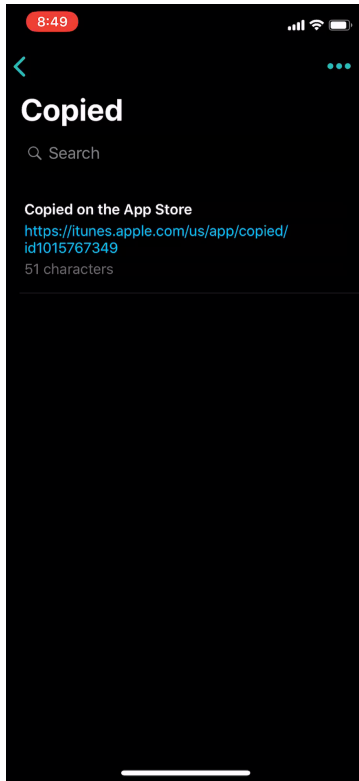
ACCIONES PARA ENVIAR UN TEXTO A COPIED

Luego se preparará la URL con la configuración correcta siguiendo el esquema URL que se explica en la ayuda de Copied que puedes encontrar también [en esta web](#).

```
copied://x-callback-url/save?  
title=Prueba&text=Solicitar entrada&x-  
success=shortcuts://
```

Puedes observar como en el parámetro "text" se ha introducido "Solicitar entrada" por lo que el texto que se enviará será el introducido por el usuario en el cuadro de diálogo.

Ahora usa la acción "Abrir direcciones URL" para poner en marcha la URL y verás qué ocurre en el video siguiente:



DESCÁRGALO AQUÍ

Ahora se va a hacer el paso inverso, que consiste en recuperar un elemento de la lista de Copied para que lo ponga en el portapapeles y puedas hacer uso de él.

Para ello de nuevo se prepara la url siguiendo las indicaciones de la documentación de Copied:

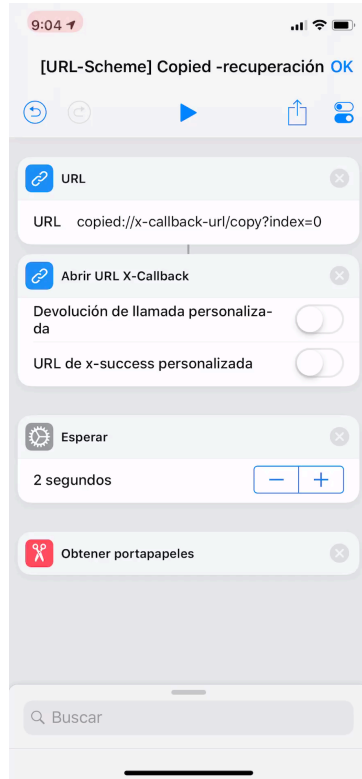
`copied://x-callback-url/copy?index=0`

El valor que puedes dejar a la elección de tu usuario en este caso es el del índice del elemento que se va a copiar, en este caso se ha puesto el 0 para que sea el primer elemento de la lista de Copied.

Ahora se utilizará la acción “Abrir URL X-Callback” para que puedas comprobar como el uso es idéntico. No se hará uso de ninguna personalización pues no es necesario en este caso.

El funcionamiento de las URL-Schemes es bastante rápido y en el caso de las X-Callback URL a menudo se vuelve tan rápido a la app desde la que se lanza que no da tiempo a que la app llamada ejecute su acción. Es por esto que tras la llamada, antes de recuperar el contenido del portapapeles se ha colocado una acción “Esperar” con 2 segundos. De esta manera se le da tiempo a Copied a poner en el portapapeles la información antes de recuperarla mediante la acción “Obtener portapapeles”.

A continuación puedes ver el vídeo del funcionamiento:



CONFIGURACIÓN DEL SHORTCUT PARA RECUPERAR UN VALOR DE COPIED



[DESCÁRGALO AQUÍ](#)

3.4.3 Ejemplo URL-Scheme Telegram

Ya son muchos los usuarios de Telegram y además esta aplicación de mensajería es el centro del Grupo de Shortcuts en Español que fue, además el lugar dónde se gestó este libro.

La relación de Telegram con las URL-Schemes siempre ha sido tímida, pero aún así se van a hacer un ejemplo de cómo aprovechar lo poco que hay.

```
tg://resolve?domain=[usuariodetelegram]
```

Mediante esta URL se puede abrir un chat con un usuario o un grupo concreto siempre que sepas su alias, por ejemplo:

```
tg://resolve?domain=shortcuts_es
```

Abre Telegram y prepara todo para enviar un mensaje al grupo de Shortcuts en Español.

La otra URL-Scheme disponible es:

```
tg://msg?text=[Mensaje]&to=[telefono]
```

Que sirve para enviar un mensaje de texto a un usuario determinado siempre que sepas su número de teléfono que deberás escribir en formato internacional, en el caso de España: con +34666123456

Esto no sirve para grupos, pues no tienen asignado un número de teléfono.

```
tg://msg_url?url=[WEB]&text=[Texto]
```

Con esta URL-Scheme se abrirá telegram y aparecerá un cuadro de dialogo para seleccionar a qué usuario le enviarás directamente un mensaje con una URL y un comentario. Por ejemplo:

```
tg://msg\_url?url=http://www.shrtcts.es&text=Esto%20libro%20tienes%20que%20leerlo
```

Observa que el texto debe estar codificado como URL.

```
tg://msg_url?url=[WEB]&text=[Texto]&to=[teléfono]
```

Si a la url anterior le añades el parámetro “to” con el número de la persona a la que lo quieres enviar, directamente aparecerá el mensaje listo para enviar.

Como ves, no hay mucho que se pueda hacer con Telegram, por ahora. Pero con un par de ejemplos aprenderás a sacarle todo el partido a lo poco que hay.

Ejemplo 1. Enviar el resultado de un shortcut a un usuario de telegram

Este shortcut será un sub-shortcut al que se llamará desde otros shortcuts para hacer el envío a telegram de la información que se reciba, por tanto, se preparará para trabajar con la información que entre.

Lo primero es configurarlo como “Extensión” para ello deberás abrir la configuración del shortcut y seleccionar la opción “Mostrar al compartir”.

Activando esto, cuando vuelvas al lienzo en blanco verás que ahora arriba del todo pone “Este atajo acepta **Cualquiera**”. Eso significa que tal y como está configurado aceptará cualquier tipo de información. Pero como sabes que solo puedes enviar texto haciendo uso de las URL-Schemes de Telegram, es mejor que pulses sobre cualquiera y elijas el tipo de información que sea texto:

- Direcciones de correo electrónico
- Enlaces de mapas
- Números de teléfono
- Texto enriquecido
- Texto
- Direcciones URL

Con todos esos tipos el envío funcionará correctamente. Pulsa en OK.

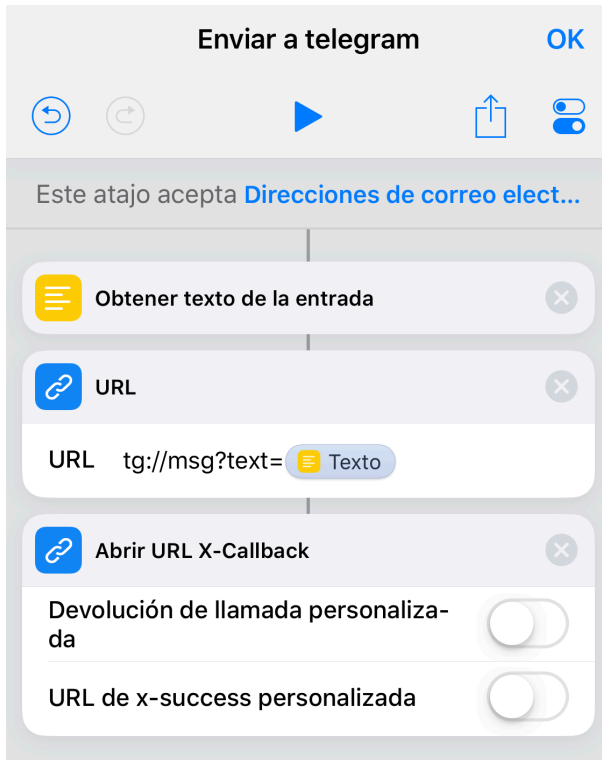
En primer lugar coloca una acción del tipo “Obtener texto de la entrada” para asegurarte de que sea lo que sea lo que entre, solo se recupere en modo texto plano, sin formato.

A continuación coloca un bloque URL y usa esta dirección

```
tg://msg?text={texto}
```

{texto} es la variable mágica que da como resultado el texto obtenido en la primera acción.

Usa, ahora, la acción “Abrir URL X-Callback” para que se abra la URL.



FLUJO COMPLETO DEL SHORTCUT

Verás como al ejecutar el shortcut se usa el texto que se haya recibido y se abrirá telegram con una pantalla para seleccionar el contacto al que le quieres enviar ese texto.



[DESCÁRGALO AQUÍ](#)

Ejemplo 2. Enviar una web a un grupo de Telegram

En el segundo ejemplo vas a crear un shortcut que te servirá para enviar la web que estes viendo en Safari a un grupo de telegram.

Como se ha explicado no se pueden predefinir mensajes para enviar a grupos de Telegram pues la URL que lo permite solo admite números de teléfonos, por lo que se usará un “truco” que consistirá en “copiar al portapapeles” el texto que se va a enviar y usar la URL-Scheme de telegram para abrir la app por el chat del grupo, por lo que solo te quedará pegar el texto y pulsar enviar para conseguir tu objetivo.

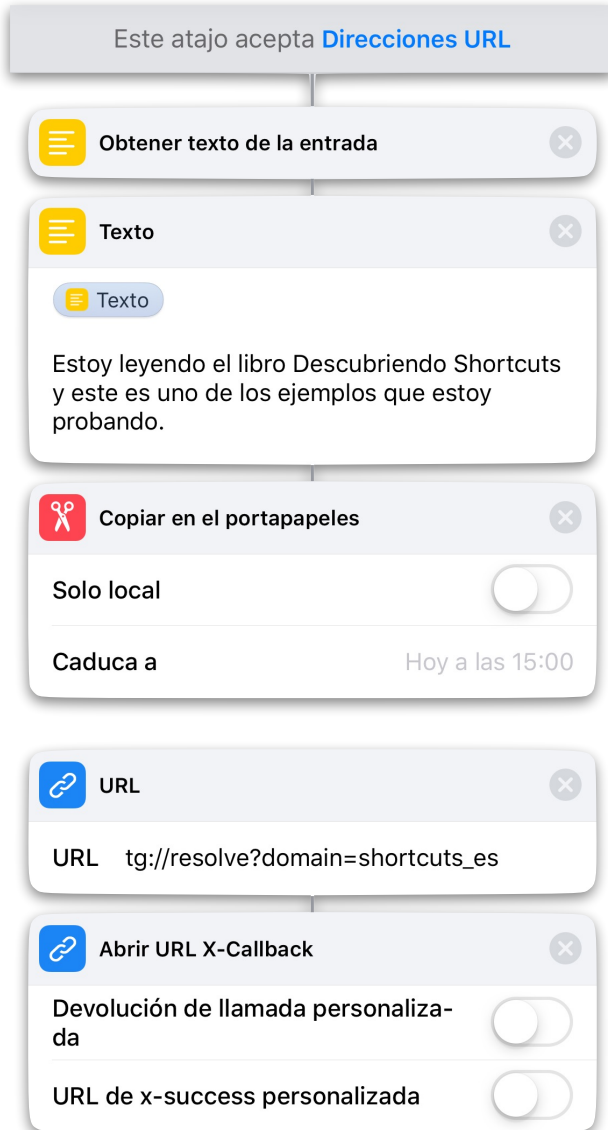
De nuevo será un shortcut del tipo compartir, y esta vez solo se seleccionará URL como tipo de contenido admisible.

Se obtiene el texto de la entrada y se copia en el portapapeles con la acción “Copiar en el portapapeles”. En el ejemplo que se ha colgado se ha añadido un pequeño huevo de pascua. Antes de copiar en el portapapeles se ha unido la url a un texto informativo sobre el libro, para que al enviarlo al grupo de telegram en español sepan que estas haciendo un ejemplo del libro.

A continuación en una acción “URL” escribe la URL-Scheme:

tg://resolve?domain=shortcuts_es

Y la acción “Abrir URL X-Callback” para que se abra la URL.



RESULTADO FINAL DEL SHORTCUT CON UN HUEVO DE PASCUA
PARA CUANDO PRUEBES EL USO DE ESTE EJEMPLO.



[DESCÁRGALO AQUÍ](#)

3.4.3 Ejemplo X-Callback URL sencillo

Los X-Callback permiten que tras llamar a una aplicación para hacer una tarea automáticamente se vuelva a Shortcuts para continuar la ejecución.

Mediante un ejemplo muy sencillo se va a hacer un shortcut que añada un vaso de agua al registro de Waterminder.

Para ello comienza pidiendo al usuario que introduzca cuantos vasos de agua se ha bebido mediante la acción “Solicitar entrada”.

Un vaso de agua tiene aproximadamente 250ml de agua, por tanto multiplica los 250ml por el número de vasos que haya escrito el usuario mediante una acción “calcular”

En una caja del tipo URL escribe la URL apropiada según la documentación de waterminder:

```
waterminder://x-callback-url/add-water?  
amount={variable mágica}
```

La variable mágica es la que devuelve el resultado de la multiplicación.



SHORTCUT COMPLETO CON LAS ACCIONES NECESARIAS

Ahora solo queda usar la acción “Abrir direcciones URL” o “Abrir URL X-Callback” para que se ejecute la petición.

Verás que en la ejecución automáticamente se cambia a Waterminder y luego se vuelve a Shortcuts.



[DESCÁRGALO AQUÍ](#)

3.4.4 Ejemplo 2 X-CallbackURL

Para cerrar este capítulo, a continuación se expone un segundo ejemplo realizado para la gestión de notas realizado íntegramente con Bear.

En la página de la aplicación se dispone de toda una documentación perfectamente explicada de todas las llamadas X-CallbackURL. Esta app es sin duda una de las que mejor aprovecha las capacidades de este tipo de llamadas, aprovechando la respuesta a las llamadas enviando distinta información.

Bearpués, implementa el protocolo x-CallbackURL, el cual permite a IOS y desarrolladores exponer distintos métodos para hacerlos disponibles para aplicaciones de terceros (en nuestro caso, Shortcuts) y poder usar esta información de la manera deseada.

El esquema de cualquier acción es algo como:

```
bear://x-callback-url/[action]?[action parameters]&[x-callback parameters]
```

Por ejemplo, la acción “open-tag” implica los siguientes **parámetros**:

- **name**(requerido): nombre del tag
- **token**(opcional): token de la aplicación (Bear App genera un token que se puede ver en las opciones de la app)

Si la llamada es exitosa, devuelve la siguiente información en **x-success**:

Un array en formato JSON con los datos siguientes:

```
[{ title, identifier, modificationDate, creationDate, pin }, ...]
```

Es decir, desde Shortcuts podemos construir una llamada de este tipo, y al realizarla se devolverá información por parte de Bear App con un fichero en formato JSON que podrá ser tratado usando las acciones relativas a Diccionarios.

En Shortcuts puedes encontrar un conjunto de bloques para interactuar directamente con tus notas realizadas en Bear. Como el usuario podrás comprobar, usando las llamadas X-CallbackURL se puede incluso mejorar el uso de estas acciones ya predefinidas.

Los bloques que dispone Shortcuts para interactuar de una forma directa con Bear son:

- **Buscar en Bear**
- **Crear nota de Bear**
- **Obtener contenido de nota de Bear**
- **Crear nota de Bear de URL**
- **Abrir Nota de Bear**
- **Añadir a Nota de Bear**

Destacar alguna acción como "**Añadir a nota de Bear**", la cual permite añadir texto enriquecido, o un fichero a una nota existente, ya sea anteponiendo los datos a añadir, o posponiéndolos al texto existente.

Otra acción más que destacable es la posibilidad de convertir cualquier página web en un documento markdown con sus imágenes incrustadas como imagen (no como enlace), lo cual tiene un potencial más que interesante.

Un factor mas a tener en cuenta a favor de Bear: la app para el Apple Watch es simplemente genial, permitiendo crear una nota nueva mediante transcripción del texto hablado, o la posibilidad de ver cualquier nota y añadir texto a ella de la misma forma.

El siguiente ejemplo muestra todas estas acciones pero usadas de forma más compleja, y aprovechando la información retornada de las distintas llamadas que se realizan.



[Descárgalo aquí](#)

Recuerda introducir el valor del token en el diccionario inicial, este parámetro se encuentra en las opciones de la App Bear, General -> Token de la API.

Este ejemplo contiene las siguientes opciones:

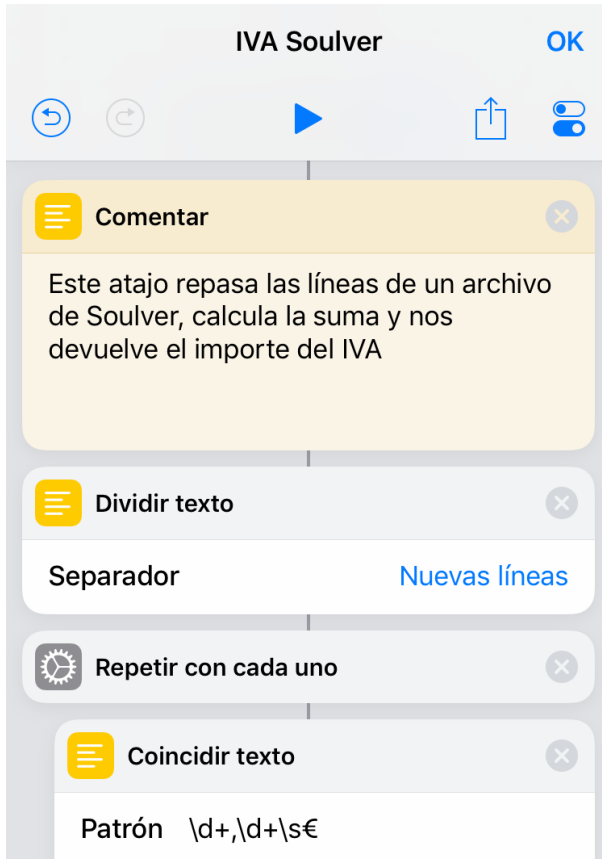
1. **Nueva Nota:** Permite al usuario dictar una nueva nota, o recuperar el texto del portapapeles para crearla. A destacar que el tecto copiado puede ser texto enriquecido con enlaces, etc...Además se muestran los tags(etiquetas) existentes, permitiendo seleccionar una o más de ellas para clasificar la nota. Finalmente se pregunta al usuario si quiere incluir algún fichero, y si la nota debe ir anclada.(si debe estar priorizada en la lista de notas)
2. **Añadir a Nota:** Permite seleccionar un tag para filtrar las notas a buscar. El usuario dicta un texto y este se añade antes o después según se escoja.
3. **Añadir Fichero a Nota:** Como su propio nombre indica, adjunta un fichero a una nota seleccionada por el usuario.
4. **Buscar Nota:** Filtra notas por una o más palabras de búsqueda.
5. **Web -> Nota de Bear:** Solicita una URL al usuario y el contenido del enlace es convertido a Markdown.

Como has podido comprobar, las capacidades de Bear desde Shortcuts son a día de hoy muy superiores a Notas, y ha servido para que puedas ver como una aplicación de IOS con una buena implementación del protocolo x-callback-url permite jugar y mucho con Shortcuts.

3.5 Comentarios

Una de las características que Shortcuts ha heredado de lenguajes de programación más avanzados es la posibilidad de insertar comentarios. Quien haya programado en alguna ocasión de seguro conoce esta herramienta tan común en entornos profesionales de desarrollo. Los comentarios son bloques de texto que acompañan al código para explicar o contextualizar el porqué de algunos de sus elementos, sin embargo, permanecen ocultos durante la ejecución pues su función se limita únicamente al entorno de desarrollo.

En Shortcuts los comentarios funcionan de la misma manera, son acciones que permiten redactar un texto con el que añadir información sobre el funcionamiento del shortcut o de alguno de sus partes, pero no realizan ninguna transformación sobre los datos.



UN EJEMPLO DE COMENTARIO EN SHORTCUTS, DONDE SE EXPLICA QUÉ HACE UN DETERMINADO SHORTCUT

El uso de comentarios es especialmente interesante cuando un shortcut es complejo y queremos recordar qué hace en cada una de sus fases, o cuando lo compartimos con otros usuarios que no han participado de su construcción.

4. Variables

En cualquier lenguaje de programación las variables son un sujeto vital, y en Shortcuts, aunque no es un lenguaje por si mismo, si que se podría definir como un lenguaje de muy alto nivel que se apoya sobre las bases de lo que se considera un lenguaje. Conocer el funcionamiento de las variables y la forma en la que se relacionan es básico para que puedas realizar con éxito tus recetas.

4.1. Shortcuts y las variables

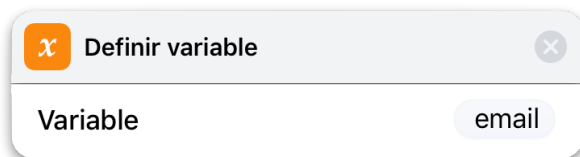
Cuando desarrolles tus primeros shortcuts, enseguida verás que necesitas recuperar valores que han aparecido en un punto anterior al que te encuentras en ese momento. Para solucionar este problema necesitas usar variables.

Una variable es una manera de almacenar un dato (texto, números, imágenes...) asignándole un nombre que sea descriptivo para que sea fácil de reconocer cuando la necesitemos, asignar un nuevo valor, etc...

A la salida de muchos bloques obtendrás datos, y estos necesitan, más adelante se verá que **no es 100% así**, ser almacenados en una variable para poder usar ese dato a posteriori. Resumiendo, dos opciones:

Asignar un valor: Para asignar un valor **resultante del bloque anterior** se usa el bloque:

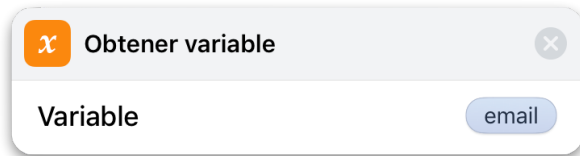
"Definir Variable + {Nombre_de_la_Variable}".



DEFINIR VARIABLE

Recuperar el valor de una variable: Para recuperar el valor de una variable se usa el bloque:

“Obtener Variable + {Nombre_de_la_Variable}”



RECUPERAR VARIABLE

A la hora de definir variables es importante tener en cuenta un par de recomendaciones, que no contengan espacios ni tilde. Un ejemplo.

- “UpperCamelCase”: La primera letra de cada palabra se pone en mayúscula. Por ejemplo : “FechaActual”
- “LowerCamelCase” Todo en minúsculas. P.Ej: “fecha_actual”

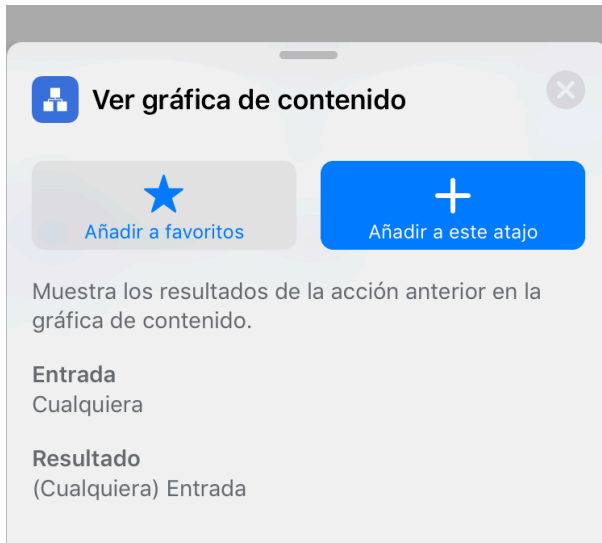
En resumen:

- “Definir Variable” recoge el valor saliente del bloque anterior y lo almacena.
- “Obtener Variable” recupera el valor almacenado, y lo entrega al bloque siguiente.

Si has coqueteado con lenguajes clásicos de programación, en estos momentos estarás esperando a que se describan los distintos tipos de variables (*String*, *Integer*, *Float*, *Boolean*), lo importante que es definir correctamente lo que se espera guardar en dicha variable, etc... Pues bien, en Shortcuts eso

no es así, una variable puede almacenar cualquier cosa. ¡Desde un texto o número, a una fotografía, un PDF, o una mezcla de todos ellos!

Es importante saber que Shortcuts hace un gran trabajo para que no se produzcan errores de tipos. Antes de explicarlo con un ejemplo, vas a aprender a usar un bloque llamado “*Ver Gráfica de Contenido*”.



BLOQUE VER GRÁFICA DE CONTENIDO

1. ¿Se puede vivir sin saber qué es la acción "Ver Gráfica de Contenido"? Sí, no es necesario.
2. ¿Por qué explicarlo? Para que se puedas entender como “piensa” Shortcuts y el alcance de las variables mágicas (4.3)

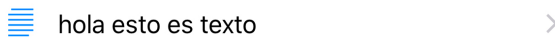
Esta clase de conversiones que realiza Shortcuts parte de la tecnología que reside en “*Ver Gráfica de Contenido*”. Debe quedar claro que para el usuario no será necesario saber cómo lo hace, pero si es importante entender que el sistema

ayuda activamente a poder ejecutar los flujos de trabajo de una forma mucho más natural y simple.



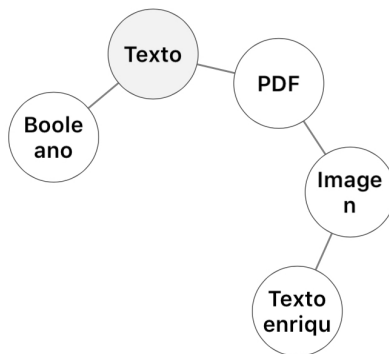
[DESCÁRGALO AQUÍ](#)

Aquí se dispone de una lista que brinda dos opciones, ambas aparentemente son texto, pero la segunda es una dirección URL. Bien, si pulsas la primera opción, verás que usando el bloque “*Ver Gráfica de Contenido*” se muestra una línea:



GRÁFICA DE CONTENIDO

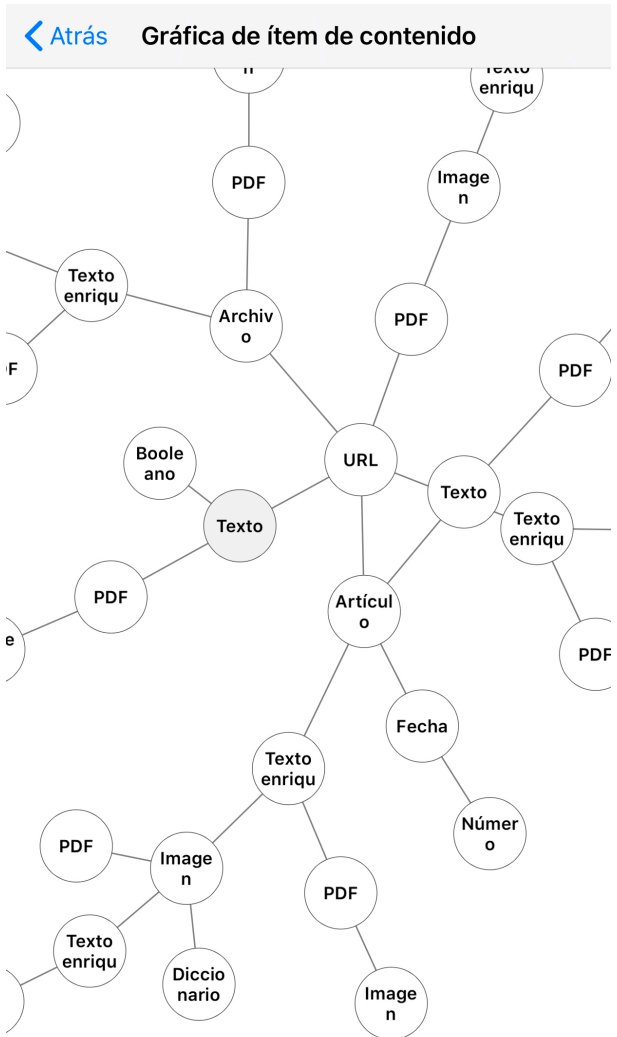
Pulsa y se abrirá a una pantalla con unos círculos llamada [Diagrama de Pert](#):



VISOR

Aquí se puede ver lo que el sistema internamente utiliza para saber en todo momento que puede hacer con cada variable. Es decir, en el caso que nos ocupa, el texto lo puede convertir a PDF, imagen y texto enriquecido.

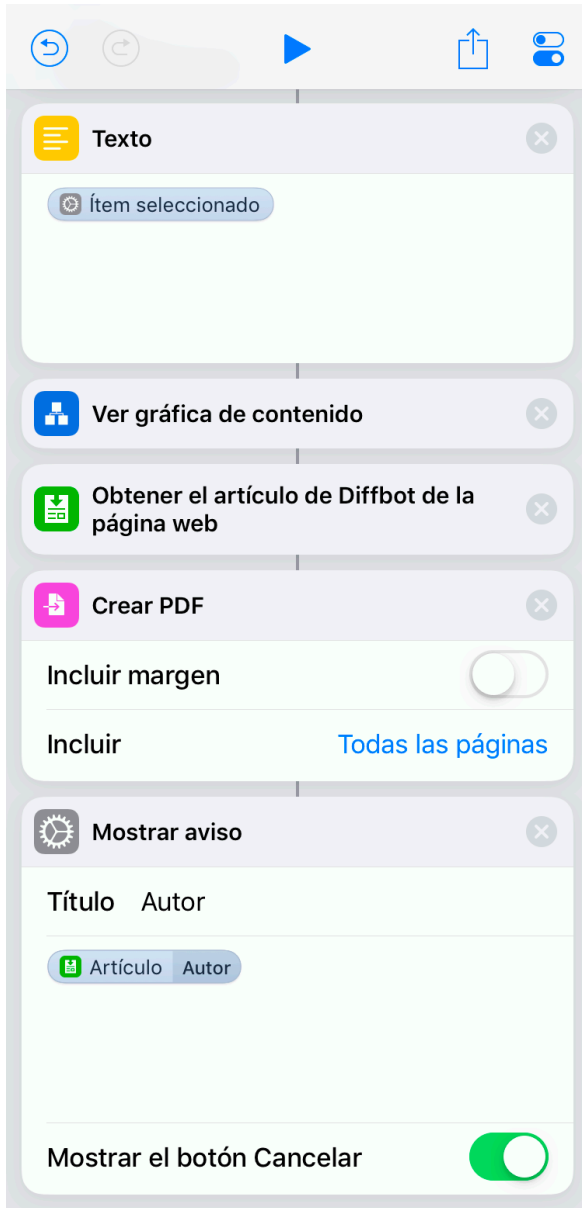
Ejecuta de nuevo el shortcut, pulsando esta vez sobre la segunda opción de la lista. Ocurre algo curioso, cuando se ve la “piscina de bolas”, se observa que aunque es un texto, el motor de Shortcuts ha leído la URL y ofrece más opciones!



GRÁFICA DE CONTENIDO DE UNA URL

En el ejemplo, sin abrir ninguna URL ni nada por el estilo se ha añadido un bloque llamado “Obtener el Artículo de Diffbot”, la app internamente sabe que debe abrir por detrás la página

y extraer el artículo. Además, como se explicará más adelante (capítulo 4.3), usando las **Variables Mágicas** no hace falta definir ningún “*Definir Variable*” para conseguir el autor de dicho artículo, lo cual ahorra bloques de ejecución y hace más fácil de seguir el flujo de trabajo. El shortcut continúa y posteriormente, con el bloque “*Crear PDF*”, Shortcuts internamente convierte lo que era un enlace URL, en un PDF completo.



CAMBIO DE TIPO

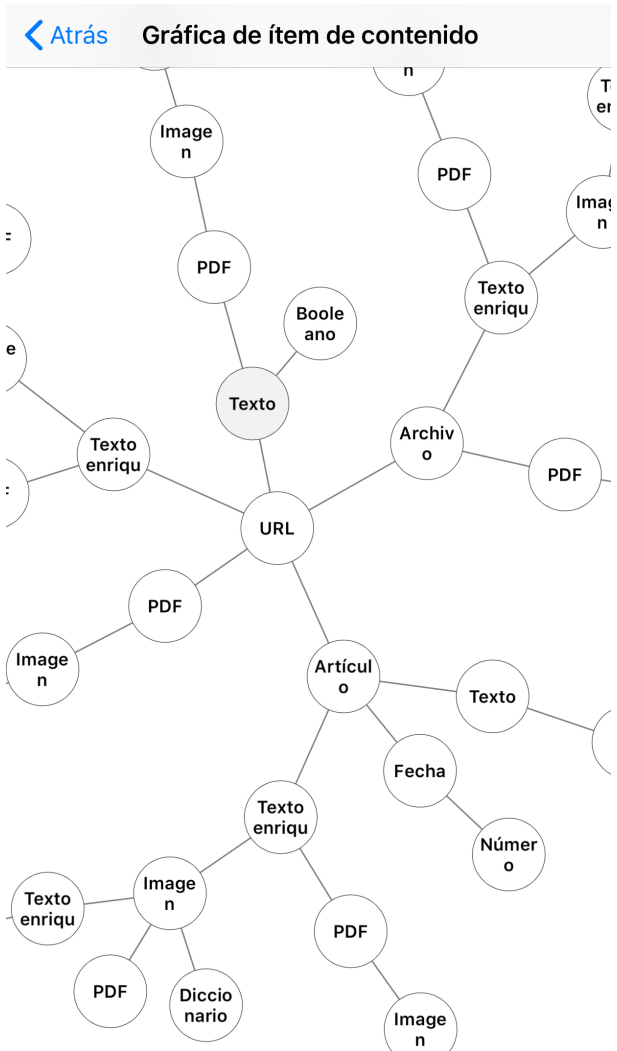
Es fantástico y a la vez sorprendente cómo Shortcuts gestiona los cambios de tipos en una misma variable, este aspecto a menudo pasa desapercibido, pero si el lector ha programado alguna vez, seguro que los errores de tipo han formado una parte importante a la hora de depurar errores. Que Shortcuts se ocupe en gran medida de esta parte evita un gran número de frustraciones para los que quieren enfrentarse a poder realizar automatizaciones, pero a su vez implica que a veces “ocurran” cosas no esperadas y es por eso que es útil tener un bloque como “*Ver Gráfica de Contenido*” para poder tener alguna pista de lo que está ocurriendo.

Otro ejemplo para entender lo que la aplicación hace sin que te des cuenta es que si se almacena una fecha y hora, y posteriormente es usada en un bloque “*Texto*”, Shortcuts la tratará como una cadena de texto (String). Por contra, si se usa esa misma variable para sumarle por ejemplo 3 días, Shortcuts cambiará, sin que el usuario deba ni tan siquiera intuirlo, el tipo de variable y la tratará como una fecha y no como texto.



[DESCÁRGALO AQUÍ](#)

En este último ejemplo, se adjunta el bloque de “*Ver Gráfica de Contenido*” y si observas en el trato de la fecha como texto ya se puede intuir que Shortcuts la “ha interpretado” y sabe que si hace falta, la puede tratar como una fecha:



GRÁFICA DE CONTENIDO DEL FLUJO

En apartados posteriores se va hacer hincapié en el modo que Shortcuts trata las variables y cómo realiza operaciones

transparentes al usuario que ahorran mucho tiempo y esfuerzo.

4.2. Definir, Obtener y Añadir

En el punto anterior ya se ha avanzado como almacenar y recuperar variables. Mira este primer ejemplo. En el siguiente shortcut se realiza una simple operación matemática entre dos números. Se puede resumir como:

- Usuario introduce primer valor -> Se almacena en la variable {Num1}
- Usuario introduce segundo valor -> Se almacena en la variable {Num2}
- Se pide que operación desea realizar.
- Se recupera {Num1} y {Num2}, se realiza la operación, y se muestra el resultado.



[DESCÁRGALO AQUÍ](#)



DEFINICIÓN DE LAS VARIABLES

En este segundo ejemplo, se va a almacenar de igual forma que en el ejemplo anterior, una fotografía, y se recuperará la variable para cambiar el tamaño. No te preocupes por ahora de nada más, simplemente debes entender que en una variable puedes guardar lo que quieras.



[DESCÁRGALO AQUÍ](#)



REDIMENSIONAR IMAGEN

En este punto de la lectura deberías ser capaz de almacenar y recuperar cualquier valor que desees. Si es así ¡sigue leyendo!

Hasta ahora sabes usar “*Definir Variable*” y “*Obtener Variable*”. Pero si te has fijado en la aplicación de Shortcuts, hay un bloque en la sección Scripts llamado “*Añadir a Variable*”. Ahora se va a complicar un poco más: imagina que en un shortcut quieres almacenar una foto, un número y un texto. Estos datos los podrías guardar en variables separadas... Pero, ¿Y si pudieras guardar estos datos tan dispares en una sola variable? Para eso se usa el bloque “*Añadir a Variable*”. Esta acción crea una lista y va añadiendo elementos del tipo que sean cada vez que se ejecuta, dentro de la variable asignada.

Shortcuts permite hacer algunas cosas que a un programador no le gustarían demasiado, ya que realiza conversiones de tipos de datos de manera autónoma y sin que tú como usuario seas consciente. Esta parte la verás más adelante, por ahora quédate con la idea.

El uso es similar a los otros dos bloques, es decir :

“*Añadir a Variable*” + {Nombre_de_Variable}



[DESCÁRGALO AQUÍ](#)



SE PUEDEN AÑADIR A UNA VARIABLE ELEMENTOS DE DISTINTOS TIPOS, NÚMEROS, IMAGENES, TEXTOS, ARCHIVOS...

Como has podido ver, la variable "ObjVariable" ha pasado de guardar un solo valor, a un conjunto de valores. Se puede entender de este modo: imagina una variable como una caja

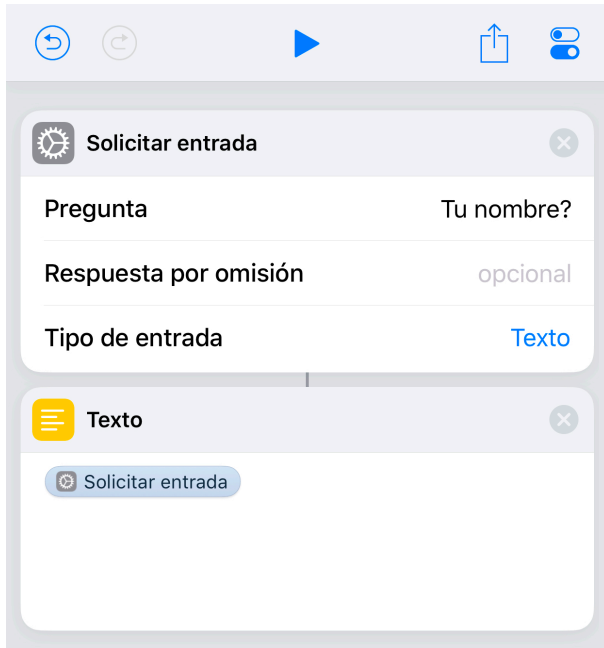
donde guardas una sola cosa (nombre) que se llama “objVariable”. Ahora, si usamos esa misma variable (“objVariable”) a la salida del bloque donde se pregunta la edad, lo que sucede es que se acaba de elevar una variable a una lista.

Es más, esta lista como se aprecia en el ejemplo ¡NO está formada por elementos del mismo tipo! Es decir, con “*Añadir a Variable*” se pasa de tener una caja con un elemento, a una estantería donde se pueden guardar varias cajas con diferentes cosas. Para acceder a una caja determinada, se usará un bloque que se detalla en el mismo atajo del ejemplo llamado “*Obtener ítem de la lista*”, o si de lo contrario, se quiere hacer algo con todos los objetos de esa lista, lo que en argot informático se llama “recorrer la lista”, se usará “*Repetir con Cada Uno*”.

4.3. Variables Mágicas

En la versión 1.7 de Workflow (9 Feb 2017) se introdujeron las “Magic Variables”, precursoras de las **Variables Mágicas** que están disponibles hoy en día en Shortcuts. Se puede definir a las variables mágicas como todo valor que se encuentra al salir de un bloque añadido a un shortcut. Es decir, si a la salida de un bloque se obtiene algún dato, ese dato es automáticamente una variable mágica. Dicho de otro modo más técnico, Shortcuts guarda en tiempo de ejecución cada valor que hay la salida de cada bloque.

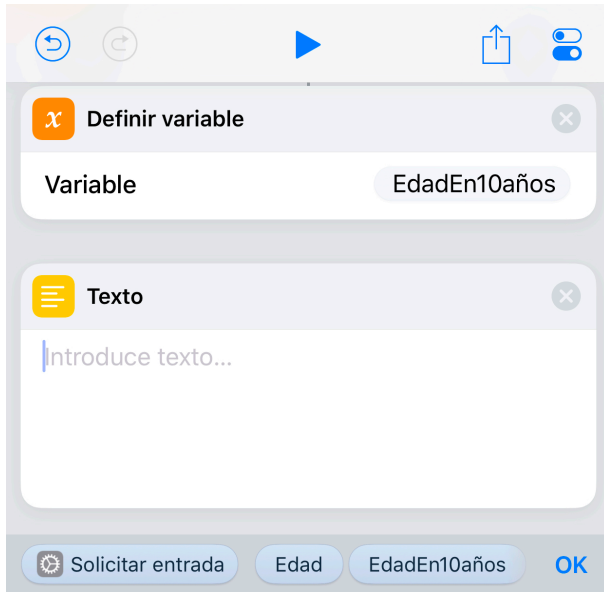
Se llaman mágicas porque las crea la aplicación, sin necesidad de que el usuario tenga que preocuparse de nada. ¿Cómo se accede a ellas? Muy sencillo, primero inspecciona el siguiente ejemplo:



VARIABLE MÁGICA DENTRO DE UNA ACCIÓN DE TEXTO

Cuando se está en una sección donde se puede escribir, el usuario verá información en una franja en la parte superior del teclado. En ella se muestran:

- Las variables definidas por el usuario (color azulado).



VARIABLES DEFINIDAS SOBRE EL TECLADO

- Las variables “especiales” (Se explicarán más adelante).
- Variables Mágicas: Son aquellas que tienen un icono adjunto y un color azulado.
- Un botón para acceder a todas las Variables definidas por el usuario.
- Un Botón con una varita para ver todas las Variables Mágicas.

Al pulsar la “varita” Shortcuts entra en un modo especial. Este permite moverse por el flujograma de tal forma que quedan ensombrecidos los bloques, apareciendo debajo de ellos variables con un icono adjunto y un color azulado. Si pulsas una de estas variables, automáticamente será añadida en el punto del programa que estemos en ese momento.



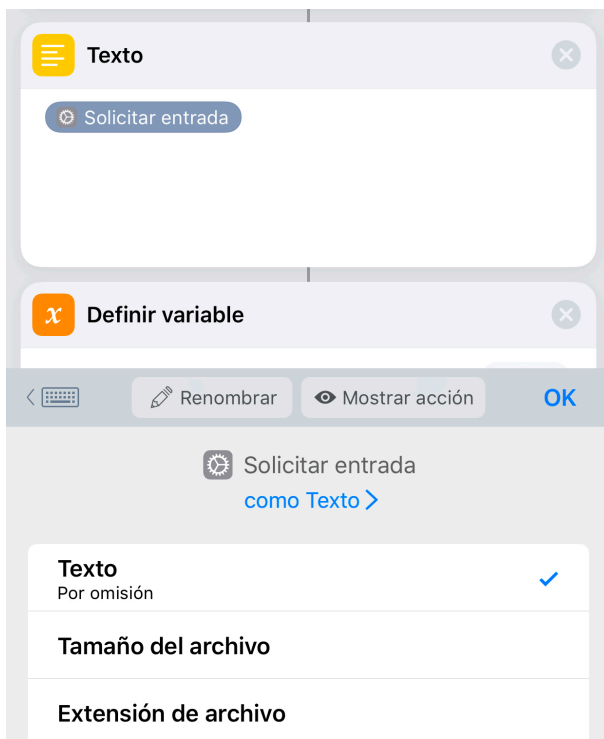
VARIABLES MÁGICAS PREPARADAS PARA SER USADAS

Una vez añadida una variable mágica a un punto de tu receta (por ejemplo en un bloque de texto), lo que verás es un nombre estándar del tipo "Texto", o "Solicitar Entrada", es

decir, nombres genéricos que harán referencia al bloque que ha generado esa variable.

Eso para un pequeño shortcut puede ser válido, pero para recetas más extensas no es suficiente. Para ello, la app brinda una funcionalidad que ofrece la opción de renombrar las variables mágicas. Esto se consigue pulsando sobre ella para que se puedan ver 2 botones:

- Renombrar
- Mostrar Acción



MODIFICANDO VARIABLES MÁGICAS

El primero permite cambiar el nombre de la variable a uno más útil y descriptivo, de tal modo que cambiará el nombre en todas las apariciones de esa variable mágica.

El segundo botón también es muy útil, ya que la aplicación realiza un desplazamiento hasta llegar al bloque que ha generado esa variable mágica, o dicho de otro modo, para que puedas saber de qué bloque proviene ese valor.

Es 100% seguro que estos dos botones pasarán a formar parte de tu rutina cuando desarrolles cualquier shortcut más o menos complejo. Mira este pequeño vídeo y observa como “aparecen las variables mágicas” y como se renombran:



¿Son imprescindibles las variables mágicas? La respuesta es no, pero te ayudarán enormemente a reducir de forma drástica tus recetas y por lo tanto, a hacer más entendible el trabajo.

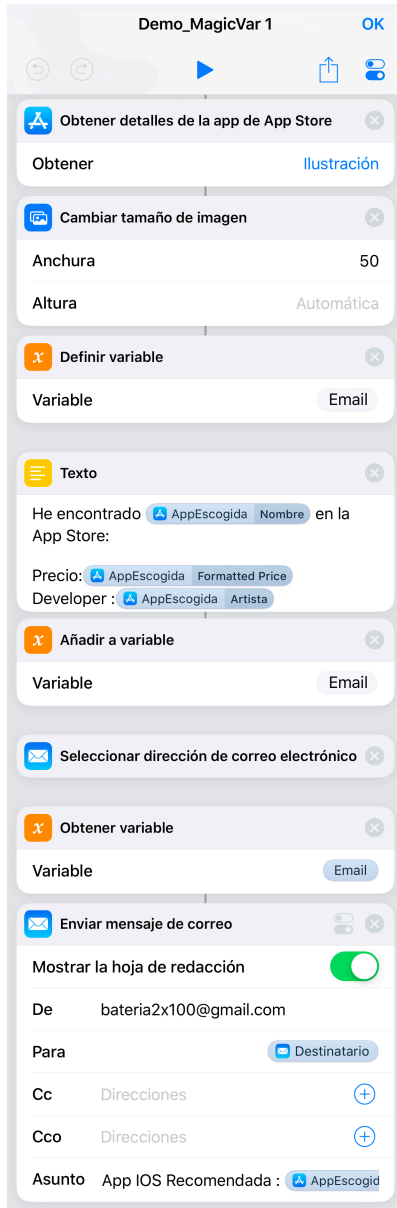
En el ejemplo [Demo MagicVar 1](#) se pueden ver dos aspectos interesantes:



[DESCÁRGALO AQUÍ](#)

1. Respecto a las Variables Mágicas: la receta busca una app en la App Store, y con el objeto que se obtiene a la salida del bloque “*Buscar en App Store*”, y sin usar ningún “*Definir Variable*”, es posible generar un mail con los datos más relevantes de la App buscada. Es decir, las Variables Mágicas permiten acceder en cualquier momento a una característica de un objeto sin apenas esfuerzo: ahorran tiempo y simplifican el shortcut.
2. En el mismo ejemplo se usa “*Añadir a Variable*” y en la lista se añaden dos objetos:
 1. El texto del email a enviar.
 2. El icono del aplicación buscada.

Un aspecto que vuelve loco de Shortcuts es que sabe interpretar que hay dos ítems en la variable (lista), y añade el texto al cuerpo del mensaje y el icono como imagen. Si fuera un PDF por ejemplo, sería un adjunto.



FLUJO COMPLETO DEL EJEMPLO

Una vez visto todo lo anterior es bueno que tengas en cuenta que Shortcuts es capaz, usando las Variables Mágicas, de realizar cambios entre tipos y extraer detalles de un objeto de una forma casi directa, de tal forma que se puede extraer la información precisa sin necesidad de realizar pasos adicionales. Es importante que se comprenda esto, ya que es transparente para el usuario, pero conocer como trabaja Shortcuts te facilitará y explicará muchas cosas.



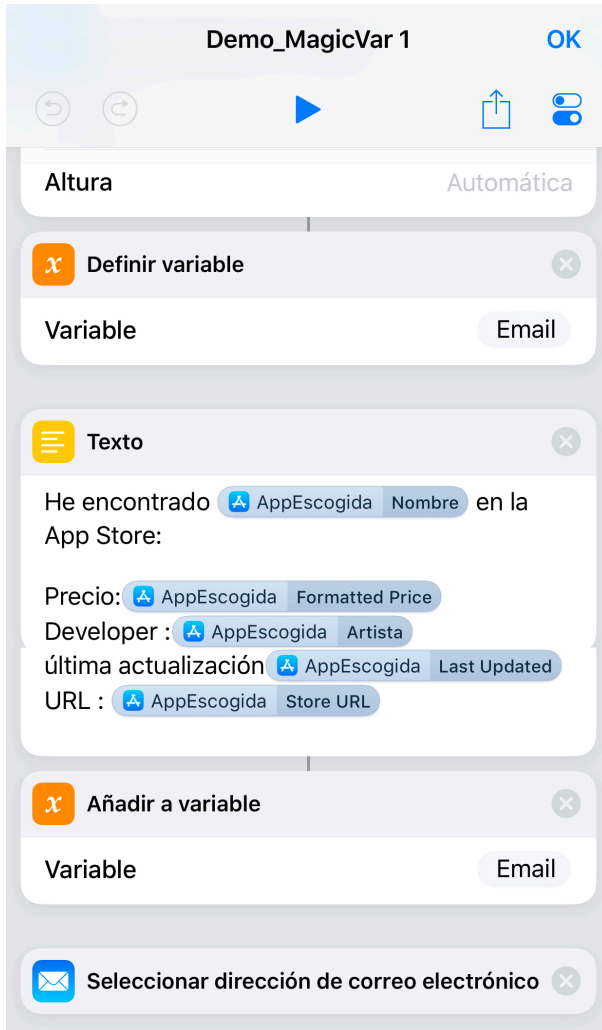
[DESCÁRGALO AQUÍ](#)



EXTRACCIÓN DE DETALLES DE MÚSICA

Si una variable mágica contiene un objeto (Ej.: un objeto aplicación fruto de una consulta en un bloque), se podrá recuperar cualquiera de las “propiedades” que contenga ese objeto. Por ejemplo, si es una App que se ha obtenido haciendo una búsqueda con el bloque “*Buscar en App Store*”, podrás recuperar cualquier propiedad del objeto “aplicación”. Esto es, el precio, fecha de publicación, categoría, etc... Todo sin tener que utilizar “*Definir Variable*”, porque Shortcuts lo hará por ti de forma transparente.

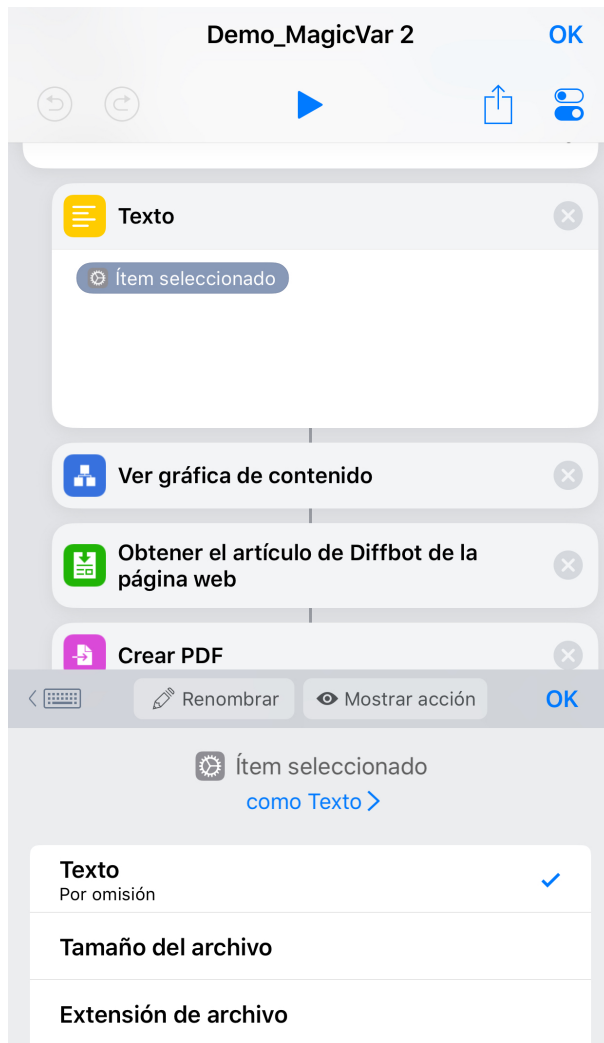
Siguiendo con el ejemplo anterior, observa como el campo “Texto” se rellena con la variable mágica AppBuscada:



RELLENAR VARIABLE MÁGICA

Hasta aquí has observado cómo las variables mágicas que contienen objetos más “complejos” permiten, de forma simple, recuperar estas propiedades que contiene el objeto padre. Pero aún hay más, al pulsar sobre una variable mágica verás

que debajo del nombre hay un texto en azul que indica “como ... >”.



VARIABLE MÁGICA CAMBIO DE TIPO



[DESCÁRGALO AQUÍ](#)

Si ahora pulsas en el texto azul se puede ver como Shortcuts ofrece un listado de cambios de Tipo. Es decir, si por ejemplo seleccionas “*Artículo*” y en el siguiente bloque se añade un “*Obtiene Detalles de un Artículo*”, y selecciona el “*Autor*”. Shortcuts, internamente, de forma transparente, recuperará la página web y la tratará como un artículo, del cual a su vez se podrá extraer detalles del tipo Autor, Fecha del artículo, etc...

Es importante que entiendas algo: no es necesario que le digas de forma manual el cambio de tipo, Shortcuts lo hará por ti. En el ejemplo, la variable mágica “Ítem Seleccionado” está como “Texto”, pero observa como Shortcuts hace el cambio de tipo y “sabe” que es una URL, de manera que en el bloque siguiente “*Obtener el artículo de Diffbot de la página web*”, interpreta la URL y descarga la página sin que tú debas hacer nada.

Siguiendo con este tipo de “magia”, observa ahora que ocurre si pasamos un link que está en un bloque de “*URL*”, y seguidamente quisieras “*Crear PDF*”. Shortcuts interpreta que no se quiere obtener un PDF con el texto de la URL, sino que recoge ese enlace y lo descarga internamente para poder realizar la conversión. La aplicación gestiona de forma automática este cambio, entendiendo que lo que quieres convertir a PDF no es la URL en si, sino su contenido.



[DESCÁRGALO AQUÍ](#)



EJEMPLO DE COMO OBTENER DATOS WEB

En este caso recupera un artículo de una web, y se recupera la URL, la fecha y el autor a partir de la variable mágica,

además de los cambios de tipo que ocurren para obtener lo que se requiere... Si no hubiera variables mágicas, implicaría para cada campo al menos 3 acciones (un “*Obtener Variable*”, “*Obtener Detalles del Artículo de diffbot*”, y un “*Definir Variable*”).

Es importante comentar una cosa más. Imagina un bloque “*Buscar iTunes Store*”, y después un “*Seleccionar de la lista*”. Recuerda, el objeto que salga del “*Seleccionar de la lista*” seguirá siendo un “iTunes product”, por lo que cuando quieras utilizar la variable mágica “*Seleccionar de la lista*” (es el nombre por defecto que tendrá la variable mágica a la salida del bloque “*Seleccionar de la lista*”), esta tendrá todas las propiedades de un “iTunes Product”, pudiendo recuperar la que necesites en ese momento, y además podrás cambiar el tipo de variable (texto, PDF, etc...) que requieras para el siguiente bloque.

Recapitulando, las variables mágicas han sido un paso muy importante para poder conseguir shortcuts centrados más en lo que se quiere hacer que en continuos bloques dedicados a “*Obtener Variable*”/“*Definir Variable*” que provocaban listados infinitos... Una característica que sin duda es imprescindible para no perder tiempo con acciones innecesarias, y centrar el foco en las tareas realmente necesarias.

El uso de “*Definir Variable*” queda relegado para ocasiones en que prima el hacer más comprensible un flujo sin olvidar que el uso abusivo de las variables mágicas complica la comprensión para otros del funcionamiento de tu shortcut, o para ti mismo, si quieres hacer una modificación pasado un tiempo.

4.4. Variables Especiales

Las variables especiales son aquellas que Shortcuts incorpora de forma automática al crear un shortcut.

Cuando se inicia un shortcut y se arrastra el primer bloque, si no es un “*Comentario*”, aparece una banda encima del teclado donde aparecen hasta 4 variables “especiales” dependiendo del contexto:

- Solicitar al Ejecutar
- Portapapeles
- Fecha Actual
- Entrada de Atajo

En detalle:

- “*Solicitar al Ejecutar*”: si se coloca, por ejemplo, en un Bloque de “*Texto*”, lo que sucede es que ese texto se “llena” con lo que el usuario teclee. Es un modo fácil de interactuar con el usuario, pero no permite ningún enunciado de ayuda como si sucede en la acción “*Solicitar Entrada*” para que el usuario entienda qué debe rellenar. Se puede aplicar en bloques como “*Repetir*”, “*Esperar*”...
- “*Portapapeles*”: Como su nombre indica, copia lo que haya en el portapapeles al bloque en que esté situada esta variable.
- “*Fecha Actual*”: La fecha actual.
- “*Entrada de Atajo*”: Asigna a ese bloque lo que haya entrado en el shortcut. Se usa a menudo cuando ese shortcut es usado como extensión. Un uso muy habitual es para usar un mismo shortcut como extensión y como shortcut independiente.



[DESCÁRGALO AQUÍ](#)

Este shortcut espera una URL, y en el caso que no tenga una URL de entrada, entiende que está en el portapapeles y realiza un PDF de ese enlace.



EJEMPLO DE VARIABLE ESPECIAL

Aquí se pueden dar dos opciones :

1. Se lanza el shortcut desde Safari, usándolo como extensión, por lo que la "Entrada de Atajo" cogerá el valor de la página de Safari en ese momento.
2. Lanzamos el shortcut desde la propia aplicación, y si tenemos en el portapapeles una URL, el resultado será un PDF de esa página.

El uso de este tipo de variables facilitan algunos pasos, son opcionales y, a menudo, ayudan a que el código sea más

claro, ahorrando algunos bloques de obtención de valores y asignación de estos a variables.

4.5. Vida de una variable

Una pregunta muy habitual en aquellas personas que consiguen realizar algún shortcut de nivel medio es: ¿Cómo puedo guardar el valor de una variable para poder usarla posteriormente?

Muy importante, las variables solo existen en el mismo shortcut, es decir, no por llamar a una variable por el mismo nombre en 2 shortcuts distintos compartirán lo que tengan almacenado.

Por definición, cuando un shortcut ha acabado su ejecución las variables que contiene se “vacían”, se pierden esos valores. Es lo que en programación se conoce como [ámbito](#) de una variable.

Pero no todo está perdido, hay maneras de mantener esos valores para que, si es necesario, recuperases un valor fuera del shortcut actual. Usando el ingenio y cómo no, iCloud, será posible que determinados valores [persistan](#) aunque haya finalizado un shortcut. También es importante adelantar que se pueden pasar datos entre shortcuts, aunque esto se explicará más adelante.

El uso de iCloud es clave para poder “alargar” la vida de una o varias variables, y es importante tener claro que los contenedores que almacenan valores son ficheros de texto.

Para poder empezar a trabajar con un ejemplo primero se aconseja descargar una de estas tres aplicaciones que facilitarán el trabajo:

- [Textor](#)
- [PreText](#)
- [Kodex](#)

Las tres aplicaciones son muy similares, y gratuitas (Pretext tiene una compra opcional para cambiar el icono y configurar *Dark Theme*)

Lo genial de estos tres editores es que son simples, y se integran perfectamente con las nuevas APIs de iOS que permiten abrir el fichero de texto en “la nube”, es decir, sin dejar una copia en tu dispositivo (“Open in place”). Destacar que Kodex ofrece para los que usen iPad, una cantidad de atajos de teclado importante.

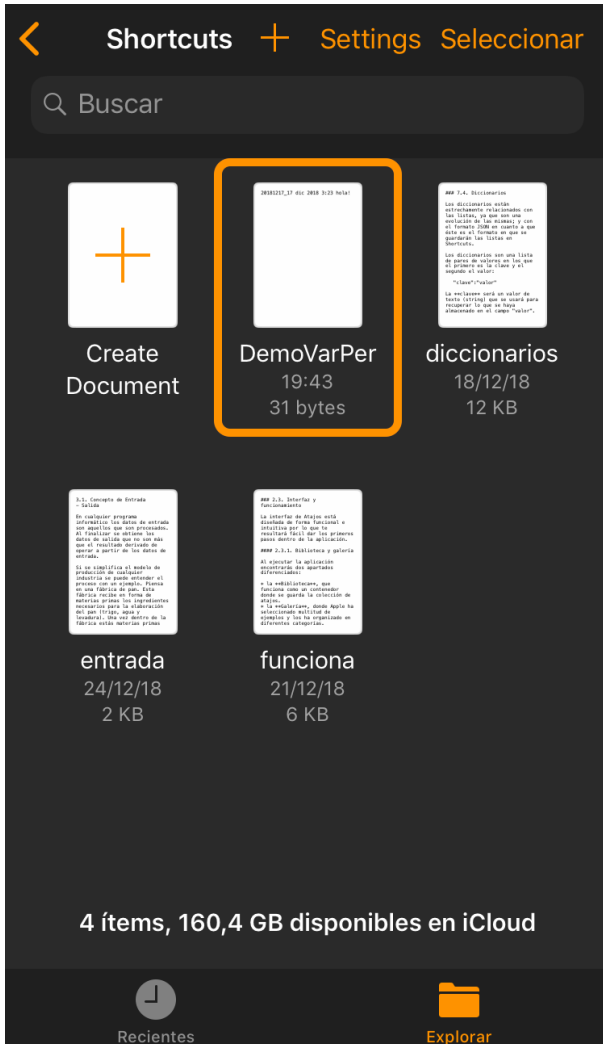
Para poder explicar de forma sencilla esta parte se va a crear un shortcut que hará lo siguiente:

1. Recuperará los archivos de la ruta iCloud/Shortcuts
2. Obtendrá una lista con los nombres de cada fichero.
3. Filtrará para buscar un Archivo que se llamará “DemoVarPer.txt”
4. Si NO existe:
 - Se generará un bloque de “*Texto*” donde se escribirá la fecha de hoy y la hora con el siguiente formato: “yyyyMMdd_HH_mm” (P.Ej. 20181231_18:35).
 - Se Guardará el archivo con el nombre “DemoVarPer.txt” en la carpeta de iCloud>Shortcuts
5. Si el fichero EXISTE:
 - Se añadirá una acción “*Obtener Archivo*” y se recuperará el fichero iCloud/Shortcuts/DemoVarPer.txt
 - Se recuperará el día y la Hora.



[DESCÁRGALO AQUÍ](#)

Si ahora abres alguna de las apps que se han comentado, podrás ver/editar el fichero creado:



FICHERO CREADO

Resumiendo, solo se creará el fichero la primera vez que se ejecute el shortcut, y partir de ese momento, las sucesivas llamadas se harán recuperando el contenido del fichero.

En este ejemplo queda claro que el formato escogido para almacenar la variable de {Fecha_Hora} es separándolos mediante un "_". Se podrían haber guardado los datos en líneas distintas, o incluso en un fichero cumpliendo formato JSON, para posteriormente recuperar la información en un diccionario.

Es importante tener en cuenta que:

- los ficheros de configuración siempre deben estar en iCloud>Shortcuts, no pueden estar en otra carpeta en el caso de usar iCloud.
- en su defecto, también se puede usar Dropbox, y en esta sí que se puede definir la ruta deseada.
- los ficheros de configuración son ficheros de texto, markdown HTML... Lo más sencillo a la hora de su tratamiento es usar ficheros JSON (se verán más adelante), pero para el ejemplo no era deseable adelantar conceptos.

Para finalizar el capítulo y resumiendo contenidos: Las variables de cualquier shortcut se borran al finalizar la ejecución de dicho shortcut. Utilizar iCloud como repositorio de ficheros de texto que contienen información es la manera más fácil para poder convertir una variable en "persistente".

5. Otros conceptos útiles

Shortcuts, como aplicación, no es independiente ni ajena a otras tecnologías o lenguajes que son bastante comunes a día de hoy. De hecho, es precisamente su capacidad para trabajar e interoperar con ellas lo que le da más potencia ya que permite salir de los pasos establecidos en la propia aplicación o en las posibilidades que otras aplicaciones introducen en la lista de acciones.

Es por esto que adquirir ciertas habilidades básicas será muy útil para llevar tus atajos a otro nivel y sacar todo el partido posible a esta herramienta.

Cada uno de los siguientes temas daría para escribir un libro, de hecho hay ya libros enteros que tratan sobre ellos, es por eso que aquí se introducirán las nociones necesarias para sacar partido a Shortcuts, pero puedes seguir profundizando una vez acabes este libro.

5.1. Markdown

[Markdown](#) es un lenguaje de marcado ligero, esto es, un lenguaje que permite dar formato a texto de manera sencilla, con poco código, y además, usando cualquier editor de texto, desde “Text Edit” en macOS hasta el “Bloc de notas” de Microsoft Windows.

Este lenguaje fue creado por John Gruber en 2004 como una manera de escribir usando texto plano fácil de leer y que pudiera ser convertido en HTML de manera sencilla.

A diferencia del HTML en Markdown no existen las etiquetas, que dificultan la lectura. Por todo esto, Markdown es, a día de hoy uno de los lenguajes de marcado más comunes. Y, de hecho, es el que se ha usado para escribir este libro.

Markdown no tiene elementos de marcado para todas las opciones que existen en HTML, de ahí que permita usarse junto con este sin problema.

Otra de las restricciones de markdown es que no tiene la opción de escribir dos líneas en blanco. Es decir, todos los párrafos estarán separados por una línea en blanco. Esto no suele ser un problema a no ser que necesites esto para algunos tipos de poemas o para conseguir algún efecto especial en tu texto.

Esta es la razón de que existan muchas variantes, también llamados sabores, de markdown que intentan, de una u otra manera, dar solución a estos aspectos. En el caso de este libro se va a explicar la versión original de markdown que es compatible al 100% con todos los programas.

Encabezados

Los encabezados permitidos son seis y se marcan con el símbolo almohadilla # al comienzo de una línea:

Título 1

Título 2

Título 3

Título 4

Título 5

Título 6

Así se establecen las categorías de títulos. Su correspondencia en html es h1, h2, h3, h4, h5 y h6.

Énfasis

Para escribir texto en negrita se requiere que se pongan dos asteriscos ****** o dos barras bajas **__** antes y después de las palabras que van en negrita, así por ejemplo "esto es ****negrita****" y "esto es **__negrita__**" será representado como "esto es **negrita**".

La cursiva sigue el mismo patrón pero se usará un solo asterisco ***** o una barra baja **_** antes y después del texto que quedará en cursiva. De esta manera "este texto ***es cursiva***" y "este texto **_es cursiva_**" se representará así: "este texto es *cursiva*".

Observa como no es necesario marcar cada palabra, se pueden introducir varias palabras dentro de los símbolos de énfasis para que todos ellos queden afectados por el formato.

Escape

Se denomina escape al carácter que permite obligar a markdown a no interpretar uno de los símbolos reservados para que aparezca como es en el texto.

Dicho de otra manera, si quieres que en tu texto aparezca un asterisco sin que el sistema lo interprete como que va a empezar una cursiva tendrás que escribir `*` esto hará que aparezca el `*` sin problema. De otra manera, todo el texto hasta que se encontrara el asterisco de cierre, el que está al final del texto en cursiva, aparecería en cursiva.

Párrafos

Para terminar un párrafo e ir al siguiente puedes darle a la tecla intro dos veces:

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam et nibh orci. Morbi ultrices eu risus et consectetur. Praesent vulputate magna libero. Etiam lobortis semper ultrices. Maecenas aliquet
```


viverra felis, eu pellentesque tellus placerat dictum. Duis sed augue in justo scelerisque luctus eget a dolor.

Vestibulum semper odio at viverra molestie. Integer ultricies sem vitae sagittis mattis. Praesent molestie lacus dui, sed elementum mauris lobortis nec. Nam laoreet sapien id tellus ultricies suscipit. Integer tincidunt dolor arcu. Vivamus eu neque vel leo mollis semper. Pellentesque hendrerit efficitur dapibus. Donec scelerisque sed lorem at vulputate.

Si lo que quieres es una nueva línea debes pones dos espacios seguidos al final de la línea.

línea 1 línea 2 produce:

línea 1
línea 2

Y

línea 1
línea 2
produce el mismo resultado:

línea 1
línea 2

Listas

En markdown están soportadas las listas numeradas, aquella que tienen números delante de cada elemento; así como las listas no numeradas, las que tienen un guión o un punto delante de cada elemento.

Listas no numeradas

Para crear una lista no numerada basta con colocar un asterisco * o un guión - delante de cada elemento separándolo en líneas:

- * Primer elemento
- * Segundo elemento
- * Tercer elemento

Producirá este resultado:

- Primer elemento
- Segundo elemento
- Tercer elemento

Y colocar un guión delante de cada elemento producirá el mismo resultado:

- Primer elemento
- Segundo elemento
- Tercer elemento

- Primer elemento
- Segundo elemento
- Tercer elemento

Así mismo se pueden crear sublistas añadiendo otra serie de cuatro espacios o un segundo tabulador para que markdown aplique la sangría correspondiente:

- Primer elemento
 - Subelemento 1
 - Sub-subelemento 1
 - Subelemento 2
- Segundo elemento
- Tercer elemento

Producirá esta salida:

- Primer elemento
 - Subelemento 1

- Sub-subelmento 1

- Subelemento 2

- Segundo elemento

- Tercer elemento

No hay límite para el número de subcategorías que puede haber.

Listas numeradas

En el caso de las listas ordenadas es igual de intuitivo, puedes colocar un número delante de cada elemento con un punto o bien colocar un 1. delante del primer elemento y * o - en los siguientes para que markdown coloque automáticamente los números siguientes de manera correlativa:

1. Primer elemento
2. Segundo elemento
3. Tercer elemento

Produce esta salida

1. Primer elemento
2. Segundo elemento
3. Tercer elemento

Igual que esto:

1. Primer elemento
- * Segundo elemento
- * Tercer elemento

Produce la misma salida que la anterior:

1. Primer elemento
2. Segundo elemento
3. Tercer elemento

Tanto es así que si se hace una lista con números pero fruto de cambiarlos de orden al rehacer la lista olvidas poner bien los números, markdown los ordenará por ti:

1. Primer elemento
7. Segundo elemento
2. Tercer elemento

Producirá esta salida:

1. Primer elemento
2. Segundo elemento
3. Tercer elemento

Así mismo en las listas numeradas pueden crearse subcategorías que se anidan bajo el elemento que las contenga. Markdown se encargará de ordenarlas y mediante CSS podrás decidir si quieres números romanos, letras o cualquier otro elemento para diferenciarlos:

1. Primer elemento
2. Segundo elemento
 1. Un elemento dentro de otro
 2. Un segundo elemento dentro de otro
3. Tercer elemento

Produce esta salida:

1. Primer elemento
2. Segundo elemento
 1. Un elemento dentro de otro
 2. Un segundo elemento dentro de otro
3. Tercer elemento

Imagina que un párrafo que escribas tiene que empezar con, por ejemplo un año, como por ejemplo:

2000. Supuso un el cambio de siglo y década al mismo tiempo.

2001. En este año se produjo el cambio de milenio.

Si lo escribes así markdown interpretará esto:

1. Supuso un el cambio de siglo y década al mismo tiempo.
2. En este año se produjo el cambio de milenio.

Muy lejos del año del que hablas. Es por eso que deberás aplicar un “escape” sobre el punto para que no piense que se trata de una lista ordenada:

2000\. Supuso un el cambio de siglo y década al mismo tiempo.

2001\. En este año se produjo el cambio de milenio.

Ahora si, markdown interpretará esto:

2000. Supuso un el cambio de siglo y década al mismo tiempo.

2001. En este año se produjo el cambio de milenio.

Enlaces

Los enlaces usan una sintaxis de corchetes y paréntesis que permite enmarcar un texto y a continuación especificar la url a la que apunta.

```
[Texto enlazado](https://dirección/a/la/web.html)
```

Este sistema que es bastante sencillo añade algo de ruido (molestias) a la hora de leer el texto cómodamente, de ahí que exista una manera alternativa de escribir enlaces.

La alternativa es enmarcar el texto con corchetes y a continuación en unos nuevos corchetes escribir una id que hará referencia al enlace en una lista ordenada de enlaces al final del documento.

[Texto enlazado][1] [Otro texto enlazado][2]

[1]:<https://dirección/a/la/web.html>

[2]:<https://dirección/a/otra/web.html>

Enlaces automáticos

Si se introduce un enlace rodeado de cuñas, como `<https://www.web.com>` markdown lo reconocerá automáticamente y mostrará la web.

En el caso de un correo electrónico del tipo `<direccion@mail.com>` markdown no solo lo reconocerá, además añadirá cierta codificación al enlace para evitar a los robots rastreadores para generar spam.

Imágenes

La sintaxis para las imágenes es similar que la de los enlaces.

```
![Descripción de la imagen](/dirección/a/la/imagen.jpg)
```

Fíjate en el signo de exclamación que hay antes del corchete, esto es lo que diferencia un enlace de una imagen.

Se puede añadir un título a la imagen de esta manera:

```
![Descripción de la imagen](/dirección/a/la/imagen.jpg "título de la imagen")
```

Citas

Un párrafo de cita puede usarse cuando se quiere escribir literalmente un texto obtenido de otro lugar, atribuyendo a su autor las palabras, o, en libros didácticos, para marcar frases importantes del texto a modo de recordatorio.

Para hacerlo solo hay que insertar un paréntesis angular de tipo *“mayor que”* antes de la frase o párrafo, de esta manera:

> Párrafo citado

Esto se verá así:

Párrafo citado

Esto se mantendrá así hasta el siguiente punto y a parte. En caso de seguir con la cita, deberás introducir otro > para que continúe.

Código

Para escribir código deberás usar la comilla simple francesa ` . Todo lo que quede entre estas comillas será tratado como código en tu documento.

Si se da el caso de que tu código incluye estas mismas comillas puedes usar dos al principio y dos al final para que no interprete la perteneciente a tu código como parte del lenguaje de marcado, es decir, "mi código tiene ` y no pasa nada" se ve como mi código tiene ` ` y no pasa nada

Así mismo, para bloques de código, basta con usar un tabulador o cuatro espacios para que la sangría se reconozca como tal.

```
    Mi código de varias líneas  
    va aquí
```

Se verá como:

```
    Mi código de varias líneas  
    va aquí
```

Igualmente si se aplica una tabulación el resultado es el mismo:

```
→Mi código de varias líneas  
va aquí
```

Se verá de la misma manera:

Mi código de varias líneas
va aquí

5.2. RegEx para principiantes

Si ya has visto algún shortcut de más nivel, seguro que has prestado atención a tres acciones llamadas “*Coincidir Texto*”, “*Obtener Grupo del texto coincidente*” y “*Reemplazar Texto*”. En el primero hay un campo llamado “Patrón”, y en él verás caracteres extraños que misteriosamente consiguen que a la salida del bloque aparezca texto filtrado.

Estas secuencias de caracteres que son capaces de extraer patrones de texto se denominan [RegEx \(Expresiones regulares\)](#). No es la finalidad de este libro entrar en detalle a explicar RegEx, pero si se intentará dar algunas nociones para que el usuario pueda lograr resultados interesantes. Las **Expresiones Regulares** son muy usadas en lenguajes de programación, ya sea para validar cadenas de texto, como para filtrar determinadas palabras de un escrito, etc...

Una expresión regular es un patrón con el que el motor de expresiones regulares intenta buscar una coincidencia en el texto de entrada. Un modelo consta de uno o más literales de carácter, operadores o estructuras.

Las expresiones regulares proporcionan un método eficaz y flexible para procesar texto. La notación utilizada de búsqueda de patrones coincidentes de las expresiones regulares permite analizar rápidamente grandes cantidades de texto para buscar patrones de caracteres específicos para :

- Validar un texto con el fin de asegurar que se corresponde con un patrón predefinido, por ejemplo, una dirección de correo electrónico, un número de teléfono, un DNI,...
- Extraer, editar, reemplazar o eliminar subcadenas de texto.

- Agregar las cadenas extraídas a una colección con el fin de generar un informe.

Un aspecto positivo de aprender Expresiones Regulares es que podrás usarlas en multitud de lenguajes: C#, JavaScript, Python, y, como no, Shortcuts. Para muchas aplicaciones que usan cadenas o analizan grandes bloques de texto, las expresiones regulares son una herramienta indispensable. RegEx es un mundo, hay páginas y páginas centradas en desarrollo que incluyen alguna referencia. Para complicar un poco más el tema, debes saber que hay varios motores de expresiones regulares, por lo que la sintaxis varía ligeramente. Eso sí, para los filtros que se usan el 99% son comunes a todos. En particular, y para tu [know-how](#), el paquete que usa Shortcuts está basado en el motor [ICU](#).

Antes de entrar en materia se van a ver unos pocos ejemplos, y para practicarlos necesitarás herramientas. Un consejo, la mejor manera de pulir un RegEx es probarlo en alguna de las opciones siguientes.

Se presentan 3 herramientas para 3 entornos distintos y así podrás decidir por ti mismo cuál se adecua más a tus necesidades.

- <https://RegEx101.com/>: Esta página sin duda es una de las mejores para analizar y probar filtros de expresiones regulares. Incluye comentarios a la derecha de la página que explica lo que se está filtrando. Además, en la parte inferior derecha dispone de una ayuda (en inglés) muy completa.(es la que se usará para generar todos los ejemplos que verás en este capítulo)

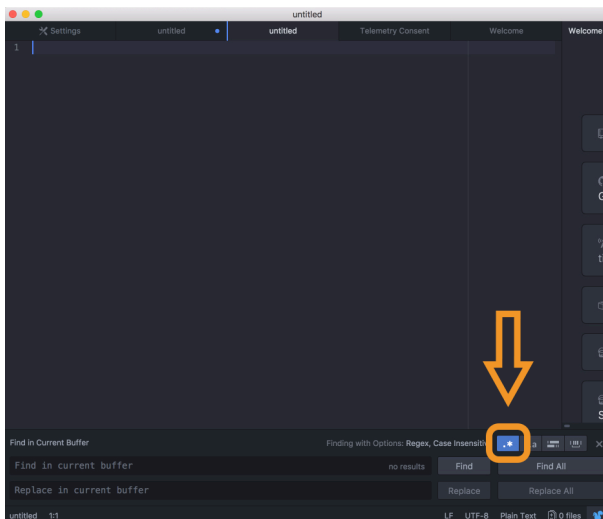
- **Aplicaciones para macOS:**

- [ATOM](#): Editor de textos gratuito. Aplicando una pequeña configuración permite probar Expresiones regulares. (explicado más adelante)
- [Patterns - The RegEx App](#) 3.49€.

- [Expressions](#). Aplicación de pago, personalmente prefiero ATOM.
- **Aplicaciones para iOS:**
 - [RegEx Knife](#) App gratuita que no está nada mal si no dispones de las otras dos opciones. Te indica los grupos filtrados (ya explicaré más adelante que es) y en movilidad, muy útil.
 - [Pretty Regular Expressions de Christopher R Hannah](#): App de pago (1.09€), algo inferior a la anterior.
 - [RegEx Lab](#) App gratuita. Funcional

Antes de proseguir, si estás usando ATOM, para usarlo como analizador de Expresiones regulares debes:

1. Abrir un documento nuevo
2. Pulsar Command + F
3. Pulsar el botón con un "."*!! que verás en la parte inferior derecha.

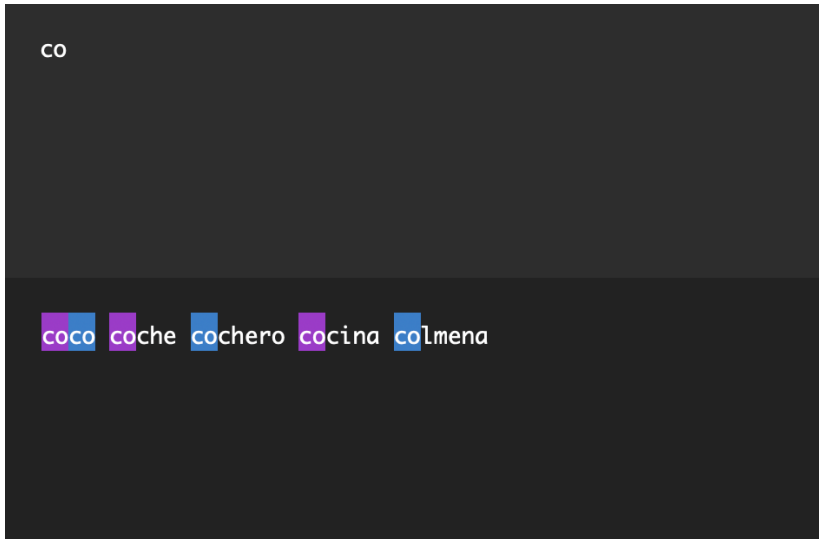


ATOM MODO REGEX

Ejemplo: Observa estas palabras

coco coche cochera cocina
colmena

Si quieres filtrar la cadena "co", lo harás de la siguiente forma:



FILTRANDO TEXTO

Como se puede intuir rápidamente, el coloreado azul indica la parte que cumple la expresión regular, en particular, hasta 6 veces se produce la coincidencia.

Mira el mismo ejemplo en shortcuts:



EXPRESIÓN REGULAR EN SHORTCUTS

El resultado a la salida del bloque “*Coincidir Texto*” es una Lista con 6 “co”. Verás con detalle las Listas en el capítulo 6.1. Es decir, cada coincidencia de la cadena “co” se almacena

como un valor en una lista, más adelante aprenderás como leer cada elemento de ella, o un elemento en particular.

En el siguiente apartado vas a ver los distintos caracteres que se usan para consultas más complejas.

5.2.1. Los Tipos de Caracteres.

Cuando uses expresiones regulares para extraer información usarás básicamente 2 tipos de caracteres :

- Literales: Como la propia palabra indica, son caracteres tal cual, sin ningún tipo de dificultad. P.Ej. “La felicidad es un sentimiento” Si deseamos buscar las “a” la expresión sería:
a
- Caracteres Especiales: El 99% del uso de RegEx se usa con este tipo de caracteres. Para realizar extracciones más complejas deberás aprender a usar este tipo de expresiones que, por otro lado, no son muy intuitivas. A estos se les llama metacaracteres, y ayudan a agregar complejidad a las consultas.

En este segundo tipo, a su vez, se pueden distinguir diferentes subgrupos (hay alguno más, pero no es objeto del libro):

- **Cuantificadores:** Un cuantificador especifica cuántas instancias del elemento anterior (que puede ser un carácter, un grupo o una clase de caracteres) debe haber en la cadena de entrada para que se encuentre una coincidencia. Son los siguientes :

: Coincide con el elemento anterior 0 o más caracteres. Ej: XYZ Concuera con cualquier cadena XY seguida por 0 o más Z. [compruébalo](#)

XYZ*

XYZ
XY
XYZZ
SXYZ

CUANTIFICADOR *

+: Coincide con el elemento anterior 1 o más caracteres. Ej:
XYZ+ Concuerta con cualquier cadena XY seguida por 1 o
más Z. [compruébalo](#)

XYZ+

XYZ

XY

XYZZ

SXYZ

CUANTIFICADOR +

?: Coincide con el elemento anterior 0 o 1 carácter. Ej: XYZ?
Concuerta con cualquier cadena XY seguida por 0 o 1 Z.
[compruébalo](#)

XYZ?

XYZ
XY
XYZZ
SXYZ

CUANTIFICADOR ?

{min, max}: Coincide entre un número de caracteres limitado por un mínimo y un máximo. Ej: $XYZ\{2,5\}$ Concuerta con cualquier cadena XY seguida por 2 a 5 Z [compruébalo](#)

XYZ{2,5}

XYZ

XY

XYZZ

SXYZ

CUANTIFICADOR MIN-MAX

{n}: coincide n caracteres. Ej: XYZ{2} Concuerta con cualquier cadena XY seguida por 2 Z [compruébalo](#)

XYZ{2}

XYZ

XY

XYZZ

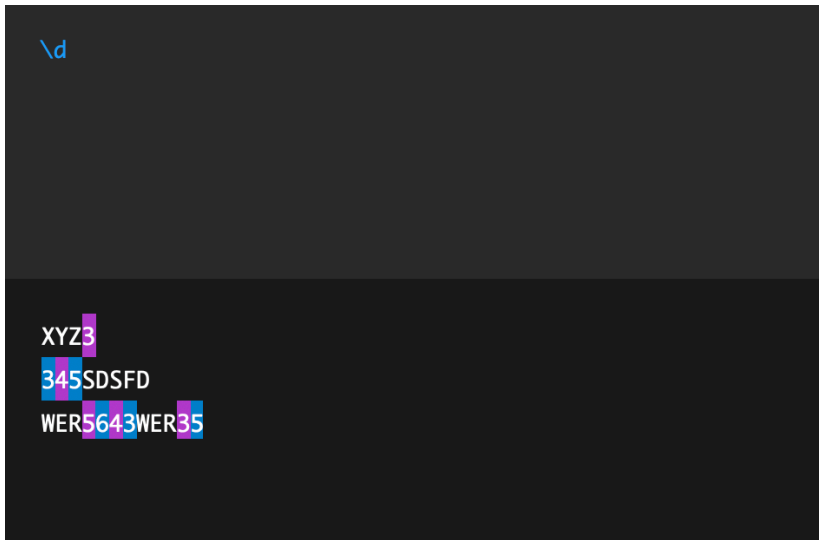
SXYZ

CUANTIFICADOR {N}

- **Clases de Carácter:** Una clase de caracteres coincide con cualquiera de un juego de caracteres. Este tipo de metacaracteres indica coincidencias con cualquier carácter de un texto. Esta clase usa de lo que se llama “carácter de escape”, que no es más que el uso de la barra diagonal inversa (\). De esta manera, se indica que el carácter que sigue no es literal, sino que el carácter que le sigue es un carácter especial.

Véanse los más importantes:

`\d` : Significa 1 dígito de 0-9. Ej: XYZ245 Concuera con cualquier carácter numérico [compruébalo](#)



DÍGITO

`\w`: Significa un carácter del rango A-Z,a-z,0-9 (Si, incluye números). Ej: XYZ245 Concuera con cualquier carácter alfanumérico. [Compruébalo](#).

```
\w
XYZ3
345SDSFD
WER5643WER35
```

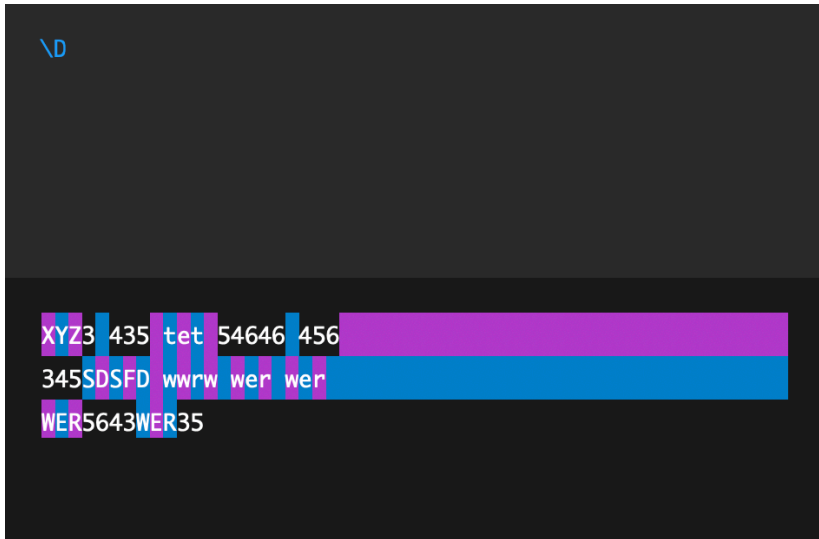
CARÁCTER

\s: Espacio en blanco. Ej: XYZ 245 Concuerta con cualquier carácter espacio en blanco, [compruébalo](#)

```
\s
XYZ3 435 tet 54646 456
345SDSFD www wer wer
WER5643WER35
```

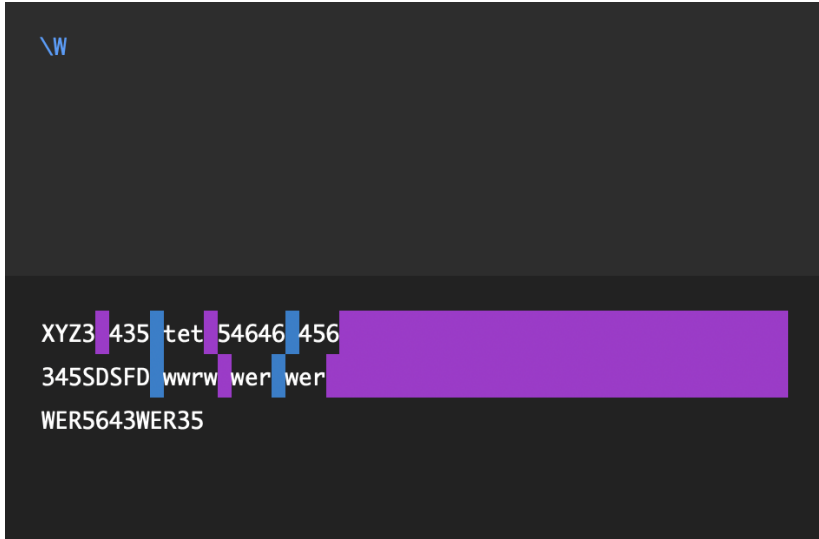

CUALQUIER CARÁCTER

\D: Cualquier carácter que no sea un dígito. Ej: [compruébalo](#)



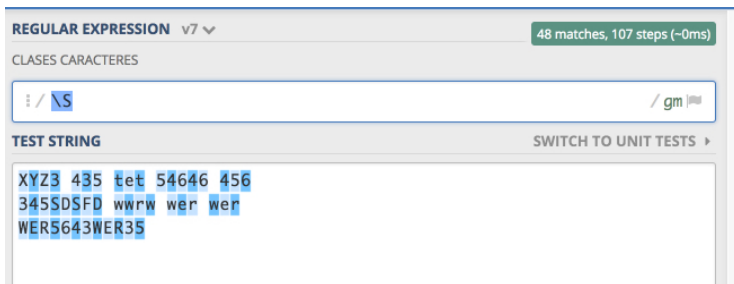
CUALQUIER CARÁCTER QUE NO SEA UN DÍGITO

\W: Cualquier carácter que no sea del rango A-Z, a-z,0-9. Ej: [compruébalo](#)



OPUESTO A

\S: Todo lo que no sea un espacio en blanco. Ej: [compruébalo](#)



OPUESTO A

- **Operador “o”** : Cláusula OR típica, o uno u otro. Dos maneras equivalentes de usarla.

I Ej. XIY. Ej: [compruébalo](#)

3 1 5

XYZ3 435 tet 54646 456
345SDSFD wwrw wer wer
WER5643WER35

OPERADOR “OR”

[] Ej. [XY] [compruébalo](#). Este operador tiene además dos opciones más a tener en cuenta cuando están dentro de los “brackets”:

- Significa “entre”. Por ejemplo: [a-z] → De la “a” a la “z”. Ej: [compruébalo](#)

[a-z]

1234 Esto es una prueba de selección de

OPERADOR “ENTRE”

^ Significa cualquier carácter que NO sea el de los brackets.

[^0-5] → Cualquier carácter que no sea de 0 a 5. - Ej:

[compruébalo](#)

[^a-z]

1234 Esto es una prueba de selección de

CARÁCTER QUE NO ESTÉ ENTRE []

- **Posición:** Los delimitadores de posición hacen que una coincidencia tenga éxito o no dependiendo de la posición actual en la cadena, pero no hacen que el motor avance por la cadena ni consuma caracteres. Se usan para “situarte” en un punto e iniciar una selección a partir de ese punto.

^: De forma predeterminada, la coincidencia debe comenzar al principio de la cadena; en el modo multilínea, debe comenzar al principio de la línea

- Ej: [compruébalo](#)

^Esto

1234 Esto es una prueba de selección de
Esto es una prueba
Prueba Esto

PRINCIPIO DE LÍNEA

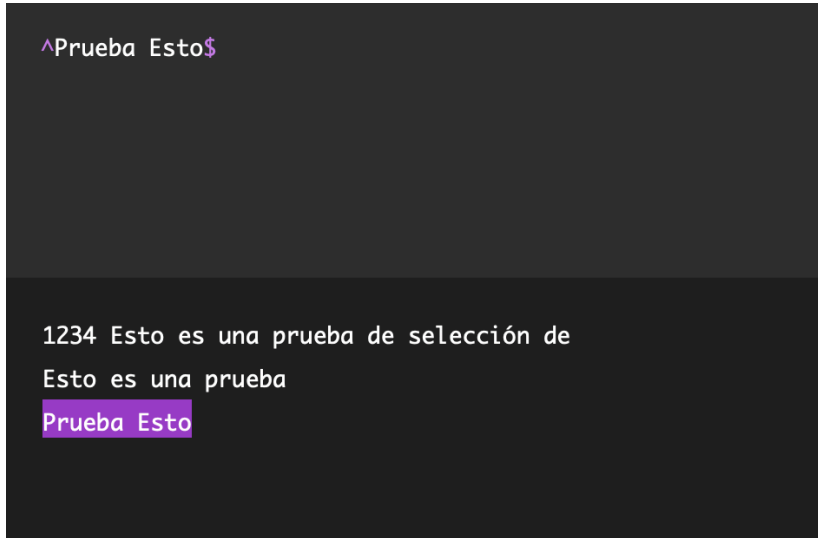
\$: Igual al anterior, pero final de línea. - Ej: [compruébalo](#)

rueba\$

1234 Esto es una prueba de selección de
Esto es una prueba
Prueba Esto

FINAL DE LÍNEA

`^Prueba Esto$` solo seleccionará aquel texto cuya línea sea estrictamente “Prueba Esto”. Ej: [compruébalo](#)



```
^Prueba Esto$

1234 Esto es una prueba de selección de
Esto es una prueba
Prueba Esto
```

EJEMPLO

`\b`: La coincidencia se debe producir en un límite entre un carácter.

`\w` (alfanumérico) y un carácter `\W` (no alfanumérico). Se usa habitualmente para seleccionar palabras (delimita entre espacios en blanco). Ej: [compruébalo](#)

```
\bpalabra\b
```

Esto es una **palabra** de prueba.

palabra de prueba

prueba de **palabra**

EJEMPLO

Antes de finalizar esta parte, un concepto más a tener en cuenta. Fíjate en este ejemplo: [Escape](#)

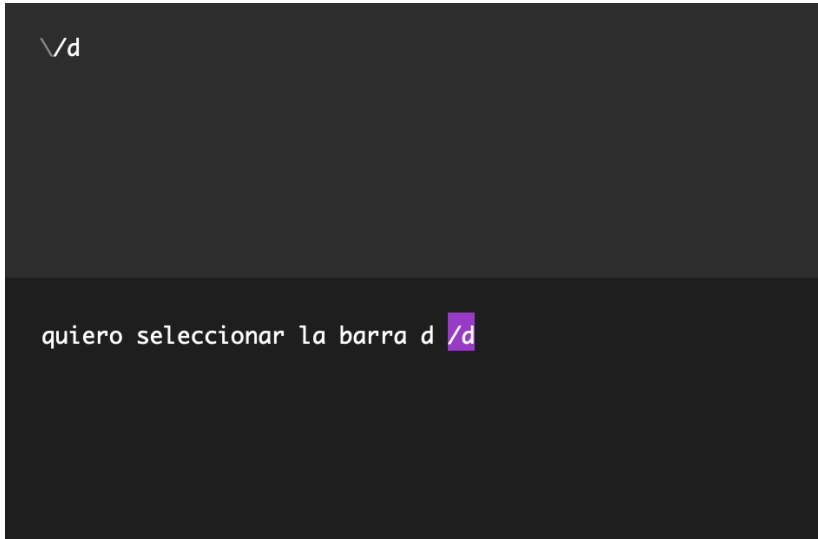
```
https://\
```

https://www.estoesunaprueba.com

http://www.estotambienesunaprueba.org

ESCAPE DE CARACTERES

Es interesante que veas cómo se “escapa” del carácter `\` mediante el símbolo `/`. De esta forma, podrás por ejemplo, realizar una consulta como [esta](#):



EJEMPLO

El “escapar” significa que el carácter como tal pierde su significado en RegEx y se toma como literal. Si buscamos `"*"` en un texto, debemos poder “escaparlos” para que no sean tomados como un cuantificador de RegEx, y eso se consigue con `/`.

Vas a seguir viendo un poco más de teoría, y al final del capítulo verás ejemplos de como usar todo esta “artillería” con Shortcuts.

5.2.2. Grupos

En RegEx los paréntesis cumplen una función muy importante, CREAR GRUPOS. Los grupos se crean mediante () y serán de gran utilidad para poder extraer coincidencias agrupadas por “tipos de coincidencia”. Enseguida verás como se crean y como se accede a un grupo en particular.

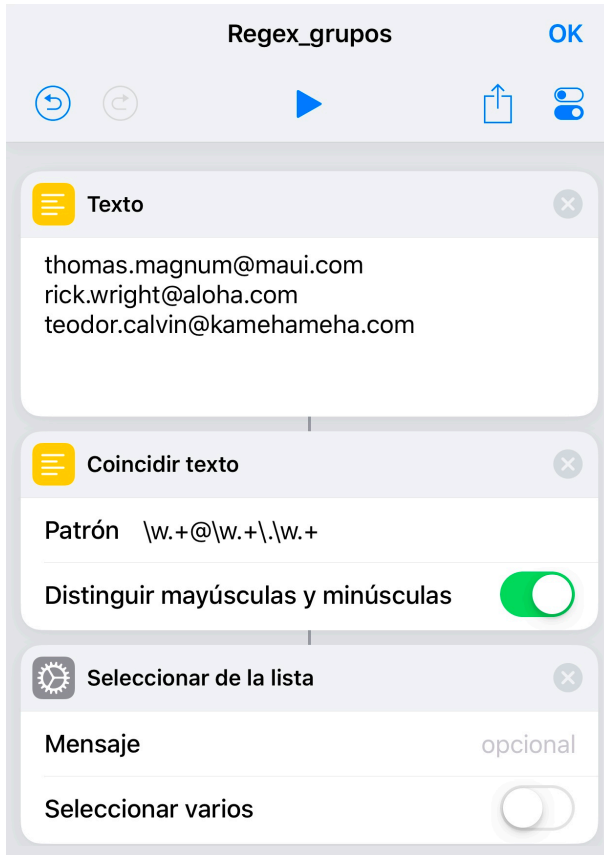
Ya has visto que, en Shortcuts, las coincidencias se devuelven en forma de lista. Mira estos emails:

```
thomas.magnum@maui.com  
rick.wright@aloha.com  
theodor.calvin@kamehameha.com
```

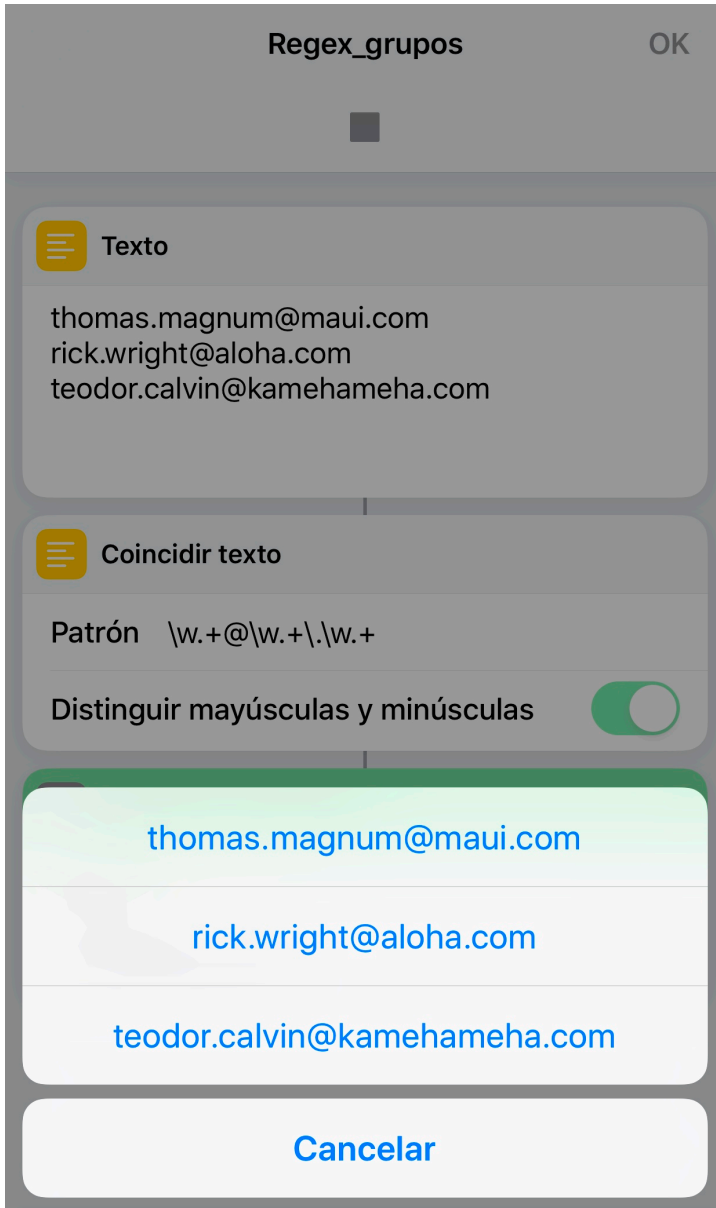
Mediante esta expresión regular se recuperan los emails:

```
\w.+@\w.+\.w.+
```

El ejemplo en Shortcuts:



GRUPOS



GRUPOS 2



DESCÁRGALO AQUÍ

Es importante remarcar que el resultado de “*Coincidir Texto*” es una lista con todas las coincidencias.






Adelanta un paso más: y si se desean obtener los diferentes dominios de los mails anteriores ? Para eso usarás los grupos: mediante el uso de paréntesis se consigue que la salida esté organizada.


- Grupo 0: Es por defecto el grupo que incluye la consulta sin paréntesis.
- Grupo 1,2...n: Los diferentes subgrupos que se forman con los (), empezando a contar de izquierda a derecha por orden de aparición de cada grupo de paréntesis.

Los grupos se nombran mediante \$0,\$1, y así sucesivamente. Mediante la acción en Shortcuts “*Obtener Grupo del Texto Coincidente*” podrás seleccionar el grupo o grupos que desees. Véase el ejemplo anterior usando agrupación:


```
\w.+@(\w.+\. \w.+)
```

Regex_grupos 1 OK


 **Texto** ✕

thomas.magnum@maui.com
rick.wright@aloha.com
teodor.calvin@kamehameha.com

 **Coincidir texto** ✕

Patrón `(\w.+)(\w.+)\w.+)`

Distinguir mayúsculas y minúsculas

 **Obtener grupo del texto coincidente** ✕

Obtener Grupo del índice Todos los grupos

Índice de grupos 2

GRUPOS 3



GRUPOS 4



[DESCÁRGALO AQUÍ](#)

Fíjate que después de obtener el grupo deseado, de nuevo se obtiene una lista, que puede ser recorrida recuperando cada uno de los dominios. De este modo podrás filtrar texto de una forma muy potente, pero a la vez compleja.

5.2.3. Coincidencias Greedy y Lazy

Los cuantificadores (* + {}) son lo que se llama operadores “greedy” (codicioso). Esto significa que “estiran” una coincidencia lo máximo que pueden a lo largo de un texto. Probablemente ahora mismo no sabes que estás leyendo. Mira este ejemplo:

```
<div> Esto es un bloque div </div>
```

Si lo que se quiere es capturar lo que hay en el interior del *div*, es decir, solo interesa “Esto es un bloque div”, una opción es la siguiente expresión regular: (<.+>)

[Ejemplo aquí](#)

(<. +>)

<div> Este bloque está dentro de un div </div>



GREEDY

Como puedes observar, el problema es que “+” extiende al máximo que puede la coincidencia, quedando un texto seleccionado que no es el deseado. Para solucionar este problema hay dos opciones, o buscar una manera distinta de filtrar el texto, o utilizar el carácter ? después del “+”:(<. +?>)



Esto provoca un cambio del comportamiento del cuantificador y lo convierte en “lazy” (vago), donde simplemente se “para” cada vez que se cumple la coincidencia, sin “estirla”:

[Greedy y Lazy 2](#)

Ahora, cada coincidencia entre <> queda de forma independiente. Automáticamente te preguntarás, ¿Y qué? A mi me interesa el texto entre <div> ...</div>. Bien, si ahora aplicas este filtro en Shortcuts usando un bloque “*Reemplazar texto*” y activando la opción “Expresión Regular”, puedes hacer algo como esto:

 Texto 

<div> Este bloque está dentro de un div </div>



 Reemplazar texto 

Buscar texto (`<.+?>`)

Reemplazar por

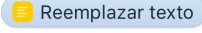
Distinguir mayúsculas y minúsculas

Expresión regular

 Mostrar aviso 

Título Alert

El texto "limpio" es :



Mostrar el botón Cancelar

EJEMPLO

NOTA IMPORTANTE: Hay un *espacio en blanco* en “Reemplazar por”.

De este modo se selecciona el texto a “eliminar”, reemplazándolo por un " " y dejando el texto sobrante como resultado. Es decir, puedes enfocar la selección de texto por

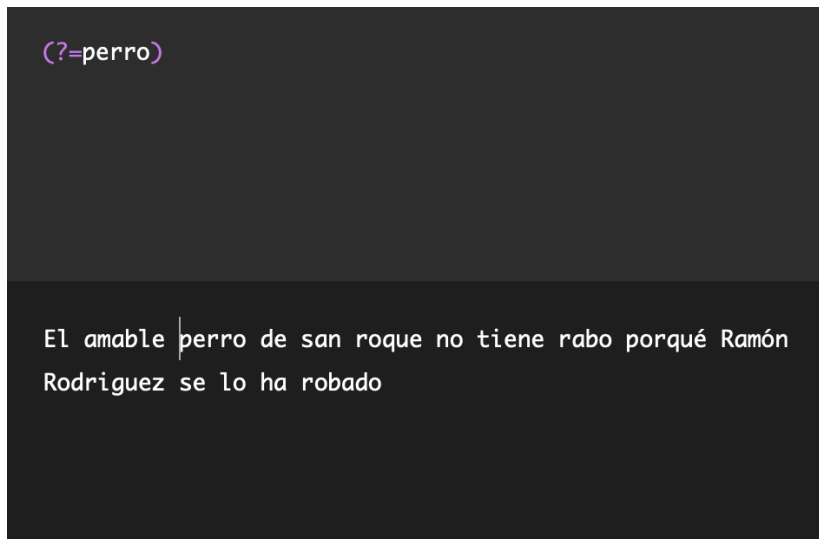
un lado, o seleccionar justamente al contrario. (El texto a eliminar).

Este tipo de técnica puede ser útil cuando realices “web scrapping”, ya que HTML se caracteriza justamente por sus etiquetas de “apertura” y “cierre” de bloques.(Ej. `<p>`y `</p>`, etc..)

5.2.4. LookAhead y LookBehind

Una última herramienta para poder enfrentarte a RegEx. La técnica de LookAhead y LookBehind sitúa el “cursor” para iniciar una selección. Se dice que no “consume”, es decir, solo se posiciona. Como en todos los casos anteriores, un ejemplo te aclarará mucho las cosas:

- LookAhead (mira hacia adelante): `(?=...)`. Observa esta imagen. Ejemplo [aquí](#)



FÍJATE EN LA POSICIÓN DEL CURSOR - LOOKAHEAD

En esta primera captura el motor no selecciona nada, simplemente sitúa el cursor antes de la coincidencia(perro),que queda “delante”

- LookBehind (mira hacia atrás): (?<=...). Ejemplo [aquí](#)

```
(?<=perro)
```

```
El amable perro|de san roque no tiene rabo porque Ramón  
Rodriguez se lo ha robado
```

EJEMPLO DE LOOKBEHIND

En esta segunda parte se sitúa el motor dejando la coincidencia después de la coincidencia (perro), que queda “detrás”.

Finalmente, mediante el uso de LookBehind y LookAhead, se acota el texto a seleccionar. Ejemplo [aquí](#)


```
(?=perro).*(?<=perro)
```

El amable **perro** de san roque no tiene rabo porque Ramón Rodríguez se lo ha robado

EJ. LOOKAHEAD LOOKBEHIND

Imagina que el objetivo fuera capturar “amable perro de san roque”: Ejemplo [aquí](#)

```
(?=amable).*(?<=roque)
```

El **amable perro de san roque** no tiene rabo porque Ramón Rodríguez se lo ha robado

EJEMPLO

Estas dos herramientas permiten situar para poder acotar más los resultados y es otra opción para poder atacar nuestras cadenas de texto.

- LookAhead: (?=...) Ej.
- LookBehind: (<=...)

Para finalizar este bloque tan duro, vas a ver algunos ejemplos aplicados a Shortcuts.

5.2.5. Ejemplos

Las técnicas de RegEx te pueden ser de utilidad para extraer datos de listas (previamente combinadas), o también, como se ha apuntado anteriormente, en Shortcuts más complejos de Captura de datos de Webs (Web Scrapping), y que se detallará en el capítulo siguiente.

Imagina un atajo en el que en un punto de la ejecución se dispone de un texto como el siguiente, en el que siempre es el mismo texto, pero cambia el número de pedido, la compañía de transportes y el número de seguimiento:

En primer lugar capturamos el pedido:

```
(\d{3}-\d{7}-\d{7})
```

(\d{3}-\d{7}-\d{7})

Pedido nº #418-1640625-0485318

La compañía de transportes responsable del envío de tu pedido es CORREOS EXPRESS. El número de seguimiento de tu pedido es 63583548596368601083900. En función del método de envío que hayas seleccionado, es posible que la información de seguimiento no se muestre de forma inmediata.

Pedido nº #121-1610615-1485118

La compañía de transportes responsable del envío de tu pedido es Amazon Logistics. El número de seguimiento de tu pedido es EQ0459678013. En función del método de envío que hayas seleccionado, es posible que la información de seguimiento no se muestre de forma inmediata.

Lo puedes comprobar [aquí](#)

A continuación, si queremos capturar la compañía de transportes:

(?<=envío de tu pedido es)(\w.*?)\.

```
(?<=envío de tu pedido es )(\w.*?)\.
```

Pedido nº #418-1640625-0485318

La compañía de transportes responsable del envío de tu pedido es **CORREOS EXPRESS**. El número de seguimiento de tu pedido es 63583548596368601083900. En función del método de envío que hayas seleccionado, es posible que la información de seguimiento no se muestre de forma inmediata.

Pedido nº #121-1610615-1485118

La compañía de transportes responsable del envío de tu pedido es **Amazon Logistics**. El número de seguimiento de tu pedido es EQ0459678013. En función del método de envío que hayas seleccionado, es posible que la información de seguimiento no se muestre de forma inmediata.

Lo puedes comprobar [aquí](#)

Finalmente, se captura el número de seguimiento:
seguimiento de tu pedido es (\w*).

seguimiento de tu pedido es (\w*).

Pedido n° #418-1640625-0485318

La compañía de transportes responsable del envío de tu pedido es CORREOS EXPRESS. El número de seguimiento de tu pedido es 63583548596368601083900. En función del método de envío que hayas seleccionado, es posible que la información de seguimiento no se muestre de forma inmediata.

Pedido n° #121-1610615-1485118

La compañía de transportes responsable del envío de tu pedido es Amazon Logistics. El número de seguimiento de tu pedido es EQ0459678013. En función del método de envío que hayas seleccionado, es posible que la información de seguimiento no se muestre de forma inmediata.

Lo puedes comprobar [aquí](#)

Otra manera de realizar este último paso, por ejemplo, sería usando LookAhead y LookBehind:

(?<=seguimiento de tu pedido es)(\w*)(?=\. En función)

```
(?<=<seguimiento de tu pedido es )(\w*)(?=\. En función)
```

Pedido nº #418-1640625-0485318

La compañía de transportes responsable del envío de tu pedido es CORREOS EXPRESS. El número de seguimiento de tu pedido es 63583548596368601083900. En función del método de envío que hayas seleccionado, es posible que la información de seguimiento no se muestre de forma inmediata.

Pedido nº #121-1610615-1485118

La compañía de transportes responsable del envío de tu pedido es Amazon Logistics. El número de seguimiento de tu pedido es FQ0459678013. En función del método de envío que hayas seleccionado, es posible que la información de seguimiento no se muestre de forma inmediata.

Lo puedes comprobar [aquí](#)

En definitiva, con 3 Expresiones regulares se puede extraer la información requerida para poder jugar con ella. En el shortcut de ejemplo se usa para añadir esta info a [Deliveries](#), [MiTrackingBot](#) y al gestor de correo de [Readle Spark](#)

Vamos a ver un último ejemplo antes de cerrar este capítulo tan árduo.

Imagina que en alguna extracción de información a través de una API (Capítulo 8) el resultado obtenido no es directamente en JSON sino que se recibe en XML.

Para el ejemplo, la info recibida es esta:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <peliculas>
    <1>
      <titulo>El Rey León</titulo>
      <year>1994</year>
      <genero>infantil</genero>
    </1>
    <2>
      <titulo>Los Chicos del Maiz</
titulo>
      <year>1984</year>
      <genero>terror</genero>
    </2>
    <3>
      <titulo>El Resplandor</titulo>
      <year>1984</year>
      <genero>terror</genero>
    </3>
    <4>
      <titulo>La semilla del diablo</
titulo>
      <year>1968</year>
      <genero>terror</genero>
    </4>
    <5>
      <titulo>Vaiana</titulo>
      <year>2016</year>
      <genero>infantil</genero>
    </5>
    <6>
      <titulo>Frozen</titulo>
      <year>2013</year>
      <genero>infantil</genero>
    </6>
    <7>
```

```
<titulo>Up!</titulo>
<year>2009</year>
<genero>infantil</genero>
</7>
</peliculas>
```

Una manera, la más habitual cuando lo realizan muchos, es empezar un listado sin fin de “*Reemplazar Texto*” de etiquetas hasta que el texto queda limpio. Gracias a RegEx y dando otro enfoque fíjate:

'<.*?>'

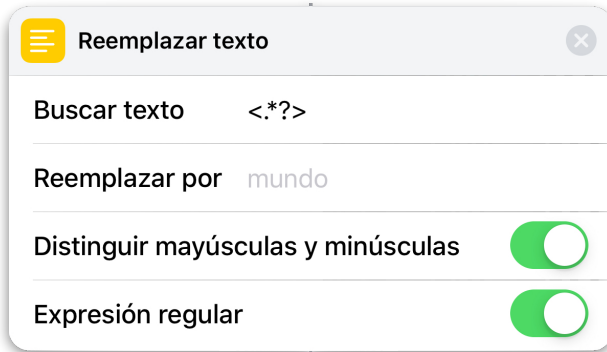
```
<.*?>

<?xml version="1.0" encoding="UTF-8" ?>
<peliculas>
  <1>
    <titulo>El Rey León</titulo>
    <year>1994</year>
    <genero>infantil</genero>
  </1>
  <2>
    <titulo>Los Chicos del Maíz</titulo>
    <year>1984</year>
    <genero>terror</genero>
  </2>
  <3>
    <titulo>El Resplandor</titulo>
    <year>1984</year>
    <genero>terror</genero>
  </3>
  <4>
    <titulo>La semilla del diablo</titulo>
    <year>1988</year>
    <genero>terror</genero>
  </4>
  <5>
    <titulo>Vaiana</titulo>
    <year>2016</year>
    <genero>infantil</genero>
  </5>
  <6>
    <titulo>Frozen</titulo>
    <year>2013</year>
    <genero>infantil</genero>
  </6>
  <7>
    <titulo>Up!</titulo>
    <year>2009</year>
    <genero>infantil</genero>
  </7>
</peliculas>
```


EJEMPLO

Lo puedes comprobar [aquí](#)

Con esta acción se consigue seleccionar todas las etiquetas, por lo tanto, esta consulta RegEx ejecutada en un bloque “*Reemplazar Texto*” de este modo:



EJEMPLO

Lo que se consigue a la salida es:

El Rey León
1994
infantil

Los Chicos del Maíz
1984
terror

El Resplandor
1984
terror

La semilla del diablo
1968
terror

Vaiana
2016
infantil

Frozen
2013
infantil

Up!
2009
infantil

Siguiente paso, eliminar estos molestos espacios:

\b\w.*\w

Alert

El Rey León

1994

infantil

Los Chicos del Maiz

1984

terror

El Resplandor

1984

terror

La semilla del diablo

1968

terror

Vaiana

2016

infantil

Frozen

2013

infantil

Up

2009

infantil

Cancelar

OK

ELIMINAR ESPACIOS

A partir de este momento, mediante un bucle y algo de imaginación, se llega a obtener una lista como esta:

El Rey León - 1994 - infantil
Los Chicos del Maiz - 1984 - terror
El Resplandor - 1984 - terror
La semilla del diablo - 1968 - terror
Vaiana - 2016 - infantil
Frozen - 2013 - infantil
Up - 2009 - infantil

RESULTADO

Usando un ejemplo real vas a ver como la curva de aprendizaje de RegEx puede casi siempre valer la pena en cuanto a claridad, ahorro de bloques, y sobre todo, poder mantener un shortcut, es decir, si cambia la fuente del formato, poder actualizar el atajo de una forma “sencilla” para que siga funcionando.

Ejemplo: Estado de SeriesTV

Usando como base los conocimientos de Web Scrapping (Capturar datos de una web abierta) adquiridos en el capítulo 5.3, vas a ver como usando RegEx se puede pasar de un shortcut de multitud de acciones, a uno de apenas 15.

En primer lugar, descarga el ejemplo sin RegEx



[DESCÁRGALO AQUÍ](#)

Si lo ejecutas, verás que funciona perfectamente, pero hay multitud de bloques de acción realmente complejos de seguir, y que requieren tiempo para poder ver dónde y porqué.

El segundo shortcut, usando RegEx



[DESCÁRGALO AQUÍ](#)

En este, usando 5 RegEx se obtiene el mismo resultado, de una manera mucho más limpia que el ejemplo anterior, donde hay numerosos bloques de “*Reemplazar Texto*”, “*Dividir Texto*”, que hacen más lento el flujo y donde cualquier cambio en la web de origen implicaría un esfuerzo en todos los bloques para poder seguir siendo funcional. En cambio, con RegEx deberías volver a sentarte delante de [RegEx101](#), pero con uno o dos cambios tendrías de nuevo tu shortcut nuevamente funcionando.

`(\r\n|\r|\n)`

[Expresión Regular](#)

`\s`

[Expresión Regular](#)

`<tbody>[<>a-zA-Z\n0-9 ()\|/í\-\&;="':. %_áÁÉÓÍÚ@ñéúó, '!' +]*<\tbody>`

[Expresión Regular](#)

```
<tr><td><strong><strong><em><strong>|
><strong>([a-zA-Z0-9 %():,ú]+)</strong></
td><td></td><td><strong>([a-zA-Z ].+?)<
```

Expresión Regular

Finalmente, se reemplazan las coincidencias con los grupos que interesan, en este caso el 2 y el 4:

MATCH INFORMATION		
Match 1		
Full match	1386-1481	`<tr><td>Aber Bergen</td><td></td><td>Temporada 3 - Estreno otoño 2019</td></tr>`
Group 1.	1393-1402	`>`
Group 2.	1402-1413	`Aber Bergen`
Group 3.	1413-1448	`</td><td></td><td>`
Group 4.	1448-1480	`Temporada 3 - Estreno otoño 2019`
Match 2		
Full match	1536-1638	`<tr><td>Absentia</td><td></td><td>Temporada 2 de 10 capítulos - Estreno 2019</td></tr>`
Group 1.	1543-1552	`>`
Group 2.	1552-1560	`Absentia`

GRUPOS OBTENIDOS

Observa que el primer RegEx es el explicado en el capítulo 5.2.1, y por otro lado, recuerda que “*Reemplazar Texto*” contiene la opción de ser usado como expresión regular (RegEx) y que se hace uso de los Grupos (\$2y \$4) para recuperar las coincidencias que deseamos.

Para cerrar el capítulo: paciencia, juega con RegEx, experimenta, fracasa... para al final poder llegar a un nivel que te permita conseguir resultados. Es bueno saber que el aprender este tipo de herramientas te puede servir para cualquier lenguaje de programación, es decir, no será un tiempo usado en vano.

5.3. Capturar datos de una web abierta

La mejor manera de obtener datos de una web siempre es haciendo uso de su API (Application Programming Interface, o Interfaz de Programación de Aplicaciones). Pero a menudo esa API no existe, no es pública (es decir el desarrollador la usa de manera interna y no la da a conocer) o es de pago, y para un uso esporádico no merece la pena pagar por ella.

En estos casos la alternativa es hacer [“Web Scraping”](#). Esta técnica es la misma que usan los buscadores web, como Google o Duck Duck Go, para indexar las páginas. Leen la web como si fueran un humano y obtienen la información que les interesa.

Esto, que puede sonar difícil, es bastante sencillo cuando se coge la mecánica.

5.3.1. El fundamento

El funcionamiento de una web, a nivel básico es el siguiente: hay un archivo html que contiene etiquetas que van entre los signos < > con información sobre como debe representarse la información que se muestra en la web. Estas etiquetas siempre tienen una apertura y un cierre, por ejemplo un título se expresa así <h1> El título </h1>. Como puede verse la etiqueta h1 se abre y se cierra, y la diferencia es que en el cierre siempre hay una /.

Ahora bien, en la etiqueta de apertura pueden hacerse más configuraciones que nunca estarán en la etiqueta de cierre, así, por ejemplo, un párrafo de texto se expresa así <p> Esto es un párrafo </p> pero si el programador ha querido cambiar algún atributo para este párrafo pueden aparecer en la etiqueta de apertura, como en este ejemplo: <p style="color: Blue"> Este párrafo va en azul</p> que representa un párrafo cuyo texto aparece en pantalla en color azul.

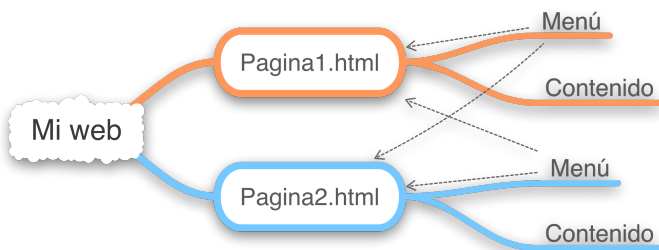
Etiquetas hay muchas y no es objeto de este libro enseñar html, pero es un lenguaje de marcado tan sumamente sencillo que en seguida irás reconociendo patrones y haciéndote con él.

El HTML, al igual que el Markdown no es un lenguaje de programación, es un lenguaje de marcado, que tal y como se explicó en el punto 1.2 sirve para dar formato a un texto.

Te sonará que existe algo llamado PHP, porque unas veces una web es del tipo `http://www.miweb.com/index.html` otras es así `http://www.miweb.com/index.php`.

PHP si es un lenguaje de programación orientado a la generación de contenido HTML de manera dinámica, es decir, mostrará una web que se crea en función de la programación interna del PHP y generará un html temporal que será el que nos muestre el navegador.

Para entenderlo fácilmente, imagina que tienes una web con un menú en el que pone *pagina 1* y *pagina 2*. Si lo quieres resolver con html necesitarás dos archivos html, uno para cada página y enlazar el menú con cada una de ellas, la estructura de mi web sería así:



ESQUEMA HTML

Puede verse como hay que repetir el menú en ambos archivos, y asegurarse manualmente de que en cada menú

que está repetido cada página html las referencias a las webs están actualizadas.



ESQUEMA PHP

En PHP solo necesitarías una página, en ella estarían los dos contenidos y el menú lo que haría sería ocultar el contenido 1 y mostrar el 2 al pulsar en contenido 2 y viceversa al pulsar en contenido 1.

Pero sea como sea, al final, lo que ves en tu navegador es un html estático, es decir no puedes ver el código PHP que ha generado ese contenido, solo puedes ver su resultado, y aún así nos podemos aprovechar de eso para nuestro propósito.

5.3.2. El procedimiento

Ahora que tienes unas nociones básicas sobre cómo funciona una web es el momento de ir aprendiendo los mecanismos para obtener información de ella.

Para este ejemplo se usará la web del [Instituto Geográfico Nacional de España](#) donde se puede ver en "tiempo real" los últimos terremotos acontecidos en este país, sus islas y Marruecos.

Esta web consta de un mapa interactivo donde se muestran los puntos de los últimos terremotos y una lista a la izquierda un listado ordenado por fecha.

Se va a crear, por tanto un shortcut que liste estos últimos terremotos para el disfrute de quién lo use. Intenta hacer el

shortcut siguiendo las indicaciones paso a paso para que te sirva de práctica, si en algún momento te pierdes, al final de este capítulo tema está el enlace para descargar el shortcut terminado.

Lo primero es echarle un vistazo al código fuente de la web de la que se va a obtener la información. Hay muchas maneras de hacerlo pero se van a describir las dos más usuales.

La primera es con tu navegador favorito. Cada uno tiene una manera distinta de ver el código fuente, pero como en este libro estás aprendiendo a usar shortcuts, vas a obtenerlo con unas sencillas acciones, que además son las primeras de este shortcut, por tanto le damos a “Crear atajo”. Pulsa en sus opciones y cámbiale el nombre a “Últimos terremotos”

Una vez dentro comienza añadiendo una acción “URL” y pega dentro la web <http://www.ign.es/web/ign/portal/ultimos-terremotos> que es sobre la que se va a trabajar.

Mediante la acción “Obtener contenido de URL” el sistema descarga la web completa al shortcut, pero intenta representarla de manera que pueda leerse, lo que no es de utilidad. Por eso usamos la acción “Convertir texto enriquecido en HTML” para que nos dé el código html que forma la web completa, este será nuestro punto de partida.

Al ejecutar el shortcut tal y como está en este momento verás como el resultado es el código fuente de la web tal y como se muestra en la imagen.

Dependiendo de la web puede que haya tanto código que sea realmente incómodo leerlo en Shortcuts. Si fuera así, como en este caso, lo mejor es copiar el texto y llevarlo a una app de texto tipo [Drafts 5](#) que es gratuita para lo que la necesitas.

Ahora tienes que buscar en el código fuente de la web aquello que te interesa para ver cómo sacarlo de ahí de una manera que nos sea útil. En este caso hay que buscar los terremotos que hay en la lista.

En este caso han hecho una tabla html cuyo tag es `<table> </table>` para listar todos los terremotos, lo que facilita la tarea.

Lo ideal es dividir los problemas a resolver en otros mas pequeños, lo que en este caso se aplica haciendo que el shortcut seleccione esta tabla y obvие el resto del código que no es de interés. Para ello usa la acción “Coincidir Texto” y tal y como se vio en el capítulo 5.2 usa un regex sencillo para seleccionar la tabla con todo su contenido usando el valor `<table>.*?</table>` ya que se ve en el código que no hay otra tabla. De esta manera nos quedamos solo con el código de la tabla.



SE SELECCIONA SOLO EL CÓDIGO DE LA TABLA

Las tablas, en html se organizan por filas, es decir, se define el comienzo y el final de la tabla con las etiquetas <table> </table> y dentro de ellas se definen las filas con <tr> </tr>. Dentro de cada fila se define cada celda de columna con <td> </td>.

Una tabla sencilla de filas y 2 columnas quedaría así:

```
<table>
  <tr>
    <td> Fila 1 - Columna 1</td><td>Fila 1
Columna 2</td>
  </tr>
  <tr>
    <td> Fila 2 - Columna 1</td><td>Fila 2
Columna 2</td>
  </tr>
  <tr>
    <td> Fila 3 - Columna 1</td><td>Fila 3
Columna 2</td>
  </tr>
</table>
```

El resultado sería este:

```
Fila 1 - Columna 1
Fila 1 Columna 2
Fila 2 - Columna 1
Fila 2 Columna 2
Fila 3 - Columna 1
Fila 3 Columna 2
```

Así que lo lógico es de la tabla obtener todas las filas en primer lugar, para ello primero vamos a limpiar el resultado eliminando al comienzo <table> <tr> y al final </tr> </table>. Usa dos cajas reemplazar texto usando el código de comienzo y final de la tabla en “Buscar texto” y dejando “Reemplazar por” en blanco como en la imagen. Nota que entre table y tr

hay un espacio y que en el primero después de `<tr>` hay un espacio y en el segundo antes de `</tr>` hay otro espacio.



SE LIMPIA UN POCO EL CÓDIGO

Ahora mismo, yendo al ejemplo de la tabla anterior tu código es este:

```
<td> Fila 1 - Columna 1</td><td>Fila 1
Columna 2</td>
</tr>
<tr>
<td> Fila 2 - Columna 1</td><td>Fila 2
Columna 2</td>
</tr>
<tr>
<td> Fila 3 - Columna 1</td><td>Fila 3
Columna 2</td>
```

La manera más sencilla de tener una lista que contenga cada fila en cada elemento es usando la acción “Dividir texto” usando como separador uno personalizado con el código `<tr>`, es decir, las etiquetas que marcan el comienzo de la cada fila. Esto da una lista de 60 elementos que son las 60 filas que hay en la tabla incluyendo la fila de cabecera.

Con objeto de evitar el uso de diccionarios ya que se explicarán en el tema 6, se van a escoger ciertos datos de utilidad y se obviarán el resto. Estos datos serán:

- Localización del terremoto
- Magnitud del evento
- Hora local del terremoto

Para evitar molestias es mejor eliminar la primera línea de la lista de filas que corresponde a la fila de título. Para ello la acción “obtener ítem de la lista” se configura así:

- Obtener: Items en el intervalo
- Obtener ítems del índice: 2
- Índice de destino: se deja en blanco para que llegue hasta el final.

Últimos terremotos

OK



Personalizar

<tr>



Obtener ítem de la lista



Obtener

Ítems en el intervalo

Obtener ítems del índice

2

Índice de destino

opcional

```
<td>es2018mgpni</td>
<td>15/10/2018</td>
<td>14:27:41</td>
<td>16:27:41</td> <td
class="tar">42.1576</td> <td
class="tar">-7.2648</td> <td
class="tar">12 </td> <td
class="tar">2.5 </td> <td
class="tac">mbLg </td> <td
class="tac">&nbsp;</td> <td
class="tal">W VILARIÑO DE
CONSO.OU</td> <td class="tac">
<form
action="http://www.ign.es/web/ign
terremotos/-/ultimos-
terremotos/getDetails"
class="form "
```



1 de 59



🔍 Buscar

AHORA LA LISTA DE ELEMENTOS HA PASADO DE SER 60 A 59

Pues bien, ahora la tarea pasa por ir escogiendo de entre las etiquetas HTML `<td>` `</td>` aquellos datos que queremos mostrar.

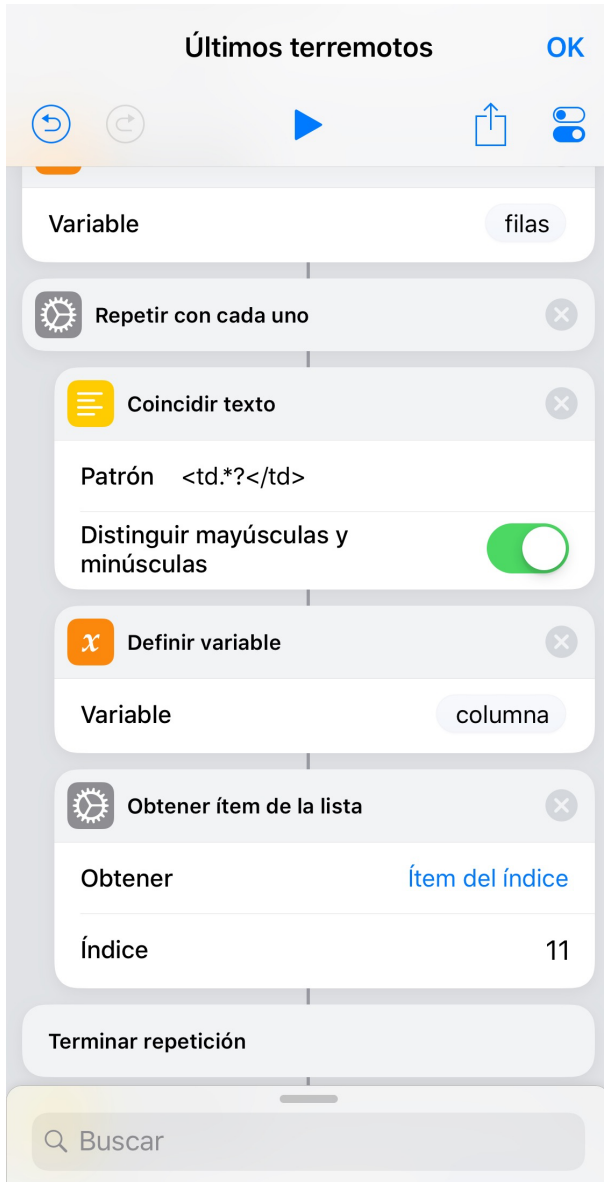
Como el flujo de trabajo requiere que se recurra a esta lista de 59 filas varias veces para ir escogiendo los valores que se desea mostrar es necesario guardarlo en una variable con la acción “Definir variable”, que se llamará `filas` ya que se han guardado ahí las filas de la tabla.

Como el filtrado hay que hacerlo para las 59 filas, en primer lugar se usará la acción “repetir con cada uno” para que lo que se aplique en una fila se aplique en todas.

Dentro de “repetir con cada uno” hay que crear una sublista de todos los elementos `<td>` `</td>` que hay en cada fila, para ello, con la acción “Coincidir texto” y el patrón `<td.*?</td>` se separarán todas las celdas en una lista. Observa que se ha elegido como comienzo `<td` y no `<td>` porque en algunos casos en el código fuente de la página se ve como esas celdas tienen configuraciones dentro. Escogiendo solo lo que es común a todas te aseguras obtener todos los datos correctos.

Pues bien, en esa nueva lista hay 12 elementos que corresponden a las 12 celdas que puede verse en la web oficial, de ellas nos interesa escoger la columna 11 que es el lugar, la 8, que es la magnitud y la 4 que es la hora local a la que se produjo.

Se guarda esa lista en una variable nueva a la que nombrarás “columna”. Y mediante la acción “Obtener ítem de la lista” con el parámetro “Obtener” en ítem del índice y el parámetro “Índice” con el número de columna en el que está el lugar, la 11, se filtrarán todas los nombres de los lugares que hay en las 62 filas.



SE SEPARAN LAS CELDAS EN UNA LISTA Y SE OBTIENE LA COLUMNA 11

Como era de esperar en esa selección viene el nombre del lugar del terremoto, y también las etiquetas HTML, por tanto se filtran para deshacerse de ellas con la acción “Reemplazar texto” usando en el campo *buscar texto* primero `<td class="tal">` y a continuación otra acción “Reemplazar texto” con el campo *buscar texto* con `</td>`

Fíjate abajo del todo como sigue poniendo **1 de 59**, esto significa que ha hecho esto en cada una de las filas y que por tanto ya se ha filtrado el nombre del lugar en todas.

Si lo vuelves a ejecutar verás como ahora los nombres aparecen correctos y sin etiquetas html. Ponemos una acción “Definir variable” y la llamamos lugar.

Ahora hay que filtrar el segundo dato que se quiere mostrar, la **magnitud del evento**. Para ello en primer lugar se recupera la variable dentro de “repetir con cada uno” mediante la acción “Obtener variable” y seleccionando la variable columnas. Y en este caso se configurará la acción “Obtener ítem de la lista” con el parámetro “Índice” en 8, que hacia referencia precisamente a la magnitud.

Igualmente deberás filtrar los datos como en ocasiones anteriores para limpiar las etiquetas HTML, en este caso eliminando la primera vez `<td class="tar">` y la segunda `</td>`. Se almacena en una acción “Definir variable” y se nombrará como magnitud.

Por tercera y última vez se repiten los pasos anteriores para obtener la hora local del evento en la columna 3, se procede de la misma manera: se recupera la variable columna, se obtiene el elemento 4 de la variable y se limpia el código html y se guarda en una variable que se llamará hora.

Con todo este trabajo hecho ahora se deben unir los datos en un texto para que los muestre de manera adecuada, para eso se usará una acción de “Texto” y dentro se usaran las

variables con los resultados de los filtros anteriores de la siguiente manera:

"lugar": "Magnitud Magnitud a las **hora"



CAJA DE TEXTO PARA MOSTRAR LOS RESULTADOS

Y ahora, para mostrar los resultados de una manera agradable a la vista, dentro de las posibilidades de Shortcuts se va a avanzar un poco en los diccionarios, aunque se verán en detalle más adelante.

Para ello al terminar la repetición se añade una acción “Combinar texto” con el “Separador” personalizado y “Personalizar” una coma ,.

A continuación se añade una acción “Texto” con el contenido de la combinación anterior, que puedes cogerla como una variable mágica pero entre llaves {Texto combinado}.



ESTE ES EL RESULTADO DEL ATAJO

Se añade la opción “Obtener diccionario de la entrada” y a continuación la acción “Seleccionar de la lista” con el mensaje Estos son los últimos seísmos registrados

Estos son los últimos sismos registrados

NW LA MONGIE.FRA

Magnitud 1.9 a las 04:30:41

SE SARREAUS.OU

Magnitud 3.3 a las 23:00:23

NE PINOS PUENTE.GR

Magnitud 2.3 a las 18:15:26

SW PLIEGO.MU

Magnitud 1.5 a las 20:26:52

SW PINOS PUENTE.GR

Magnitud 2.3 a las 01:39:27

SW LA MOJONERA.AL

Magnitud 1.9 a las 14:51:04

SW SOCUÉLLAMOS.CR

Magnitud 1.5 a las 06:53:49

S OLORON STE MARIE.FRA

Magnitud 1.7 a las 05:20:30

NE ARRAIOSLOS.POR

Magnitud 2.1 a las 23:17:05

NW FRONTERA.IHI

Magnitud 2.1 a las 05:26:15

W VILARIÑO DE CONSO.OU

Magnitud 2.5 a las 16:27:41

Cancelar

ESTE ES EL RESULTADO DEL ATAJO



DESCÁRGALO AQUÍ

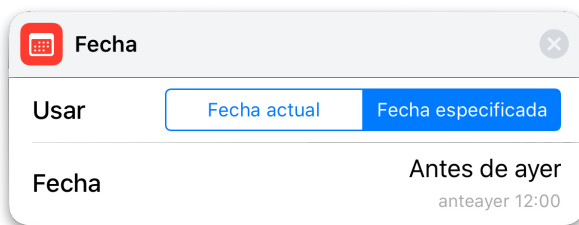
Puede que esta parte final haya sido un poco liosa por la parte del diccionario, no es mas que un truco para añadir dos líneas a la muestra de la lista final. En el capítulo de Diccionarios aprenderás más sobre ello y la potencia que tienen para tratar datos.

Como has podido comprobar aunque esta web no tuviera API, has conseguido, de alguna manera, crearla para ti. La parte engorrosa es que siempre requiere de un análisis previo del código de la web en cuestión ya que cada una será distinta. Cuantas más hagas más fácil te será descubrir los patrones para obtener los datos.

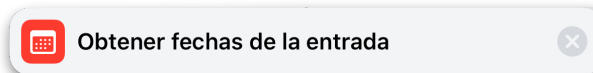
5.4. Fechas

5.4.1. Trabajando con Fechas

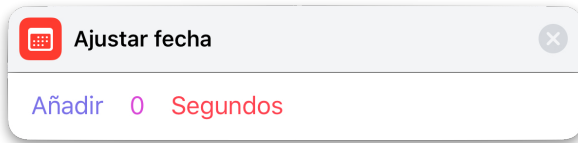
Shortcuts tiene sin duda alguna una colección de acciones lo suficientemente buena para poder trabajar con fechas. Los bloques disponibles son:



FECHA



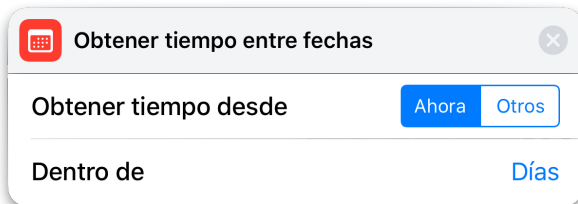
OBTENER FECHAS DE LA ENTRADA



AJUSTAR FECHA



APLICAR FORMATO A FECHA



OBTENER TIEMPO ENTRE FECHAS

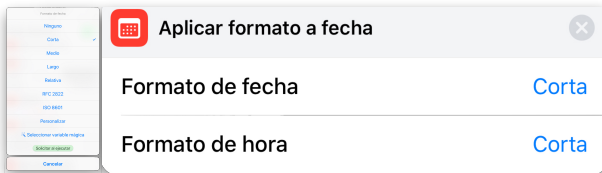
Comprobarás que el uso de estas acciones es muy común en cuanto se implementan rutinas que impliquen calendarios, búsqueda de contactos, etc.. Es decir, acciones que tienen entre sus opciones de configuración campos fecha.

Truco: El campo fecha admite lenguaje natural, donde puedes usar la fórmula por ejemplo :

Pasado mañana, Anteayer, Lunes siguiente...

Aplicar formato a fecha

En cuanto a los formatos, Shortcuts contempla los formatos definidos en [Unicode Technical Standard](#), pudiendo personalizar a un nivel muy completo el formato de fecha deseado. Para ello dispone del bloque “*Aplicar Formato a Fecha*”.



APLICAR FORMATO

Los formatos de fecha se basan en una cadena de caracteres de tal forma que cuando ese bloque reciba una fecha, a la salida entregará esa misma fecha “formateada” (es decir, modificada) al modo que el usuario haya predefinido. La acción permite configurar la fecha y la hora, por lo que se puede obtener casi cualquier combinación deseada. Los tipos más habituales son: (Tomando como ejemplo el 25 de mayo de 2016 a las 10:35)

- Corto: 25/5/16 10:35
- Medio: 25 may 10:35:00
- Largo: 25 de mayo de 2016, 10:35:00 CEST
- Relativa: Hace una aproximación respecto a la fecha actual, devolviendo para el ejemplo la cadena del tipo "2 años atrás".
- Personalizar: como el propio nombre indica, como usuario puedes personalizar el formato de salida. Es importante tener en cuenta que en el formato personalizado puedes utilizar . o / para separar días de meses, etc...

Algunos parámetros para dar formato a fechas más habituales:

- E (día): Devuelve el día de la semana, según la cantidad de E obtendrás:
 - EEE: Abreviatura: “Mié”
 - EEEE: Devuelve el nombre del día de la semana entero. Ej: “Miércoles”

w (semana): Semana del año.

- W Semana del mes
- d: día del mes
- D: Número de día del año
- yy: Formato dos cifras
 - yyyy (año): Año en formato 4 cifras
- Q (cuatrimestre): Número de cuatrimestre. Ej. 2
 - QQ: Lo mismo en formato de 2 dígitos. Ej 02
 - QQQQ: Trimestre en formato largo. Ej. 2º trimestre
- M(mes): Número de mes
- MM: En formato de dos dígitos
- MMM: Formato abreviado.Ej. Sep
- MMMM: Formato largo.Ej. Septiembre

Para horas/minutos:

- a: AM o PM
- h: Formato 12 horas.
- HH: Formato 24 horas
- m: minutos
- mm: formato de dos cifras
- s: Segundos

Entre las diferentes opciones de formato fecha seguro que habrás visto dos opciones llamadas RFC 2822 e ISO8601.

- RFC 2822: Wed, 06 May 2016, 10:35:00 +0200
- ISO 8601: 2016-05-25 Si se marca la opción "Incluir hora ISO 8601: 2016-05-25T10:35:00+02:00

RFC 2822: Tal y como indica la propia ayuda de Shortcuts, es un formato usado exclusivamente en internet y raramente será requerido.

ISO 8601: Es el formato YYYY-MM-dd, especialmente útil para renombrar ficheros y su posterior ordenación de forma alfabética, que a su vez lo es cronológica. Es decir, si tienes dos ficheros con este nombre (observa que la fecha al final del título está en formato AñoMesDía):

- foto-de-familia-20170509.png
- foto-de-mi-hijo-20170506.png

Al ordenar por orden alfabético las fechas no se tendrán en cuenta, pues es la f de familia y la m de mi las que marcarán el orden.

- 20170506-foto-de-mi-hijo.png
- 20170509-foto-de-familia.png

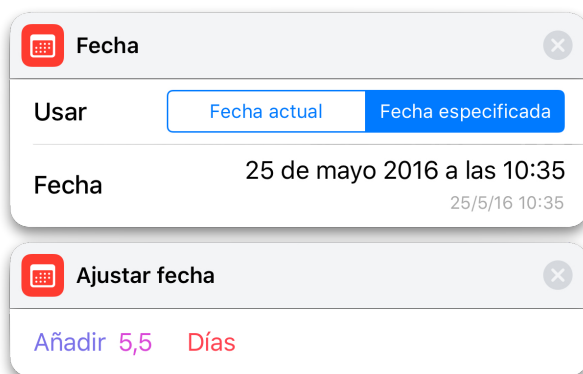
Al colocar la fecha delante del nombre, el modo alfabético de ordenación coloca las fotografías en orden cronológico. Así, sin carpetas, puedes tener todas las fotografías ordenadas por año, mes y día.

Ajustar fecha

Otra de las acciones destacables es “*Ajustar Fecha*”. Ésta te permitirá:

- Añadir/sustraer desde segundos a Años a una hipotética fecha de entrada.
- Obtener Inicio del minuto hasta Año de la fecha de entrada.
Ej. Para 26/5/2013 10:35:
 - Inicio de la hora: 10:00
 - Inicio del día: 26/05/2013 00:00
 - ...

Otro aspecto a destacar es que admite decimales para segundos, minutos, horas y días. Por ejemplo:



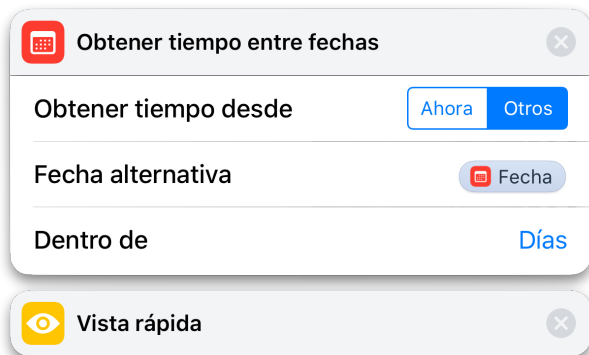
AJUSTAR FECHA CON DECIMALES

Arroja como resultado:

30 may 2016 22:35

Obtener tiempo entre fechas

Esta acción tal y como indica requiere de una fecha a la entrada del bloque, y una segunda fecha que se define en el propio bloque. (Como es lógico, una, ninguna o ambas pueden ser variables)



TIEMPO ENTRE FECHAS

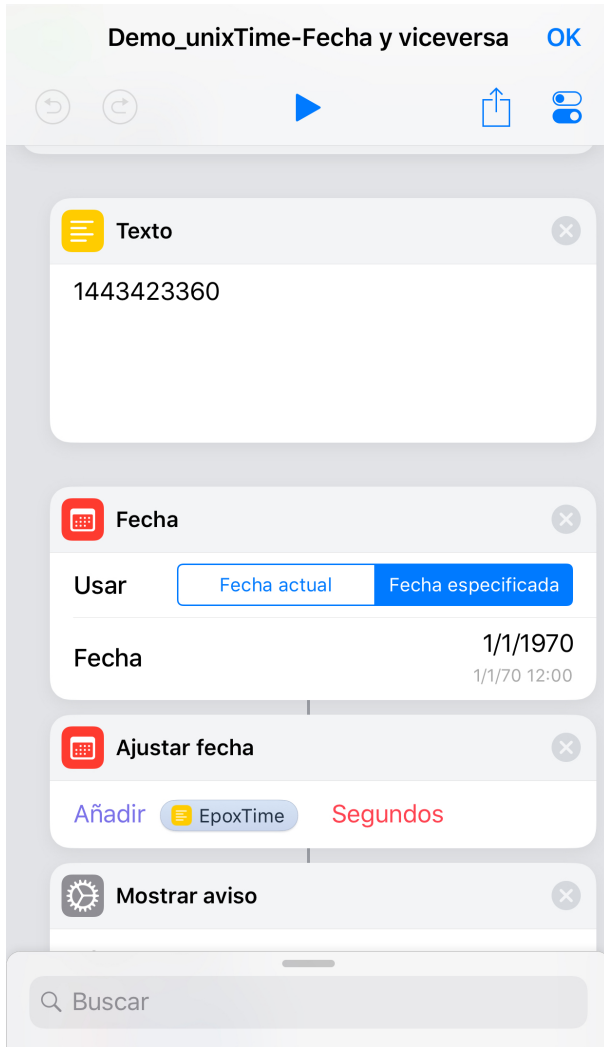
5.4.2. Unix Epoch Time

El Unix Epoch (o Unix time, o POSIX time o marca de tiempo Unix) es un modo de contar tiempo transcurrido definido como el número de segundos desde el 1 de Enero de 1970 en UTC (*Universal Time Coordinated*, es decir, lo que se llamaba meridiano de Greenwich o GMT). Este número de segundos no tiene en cuenta los segundos intercalares.

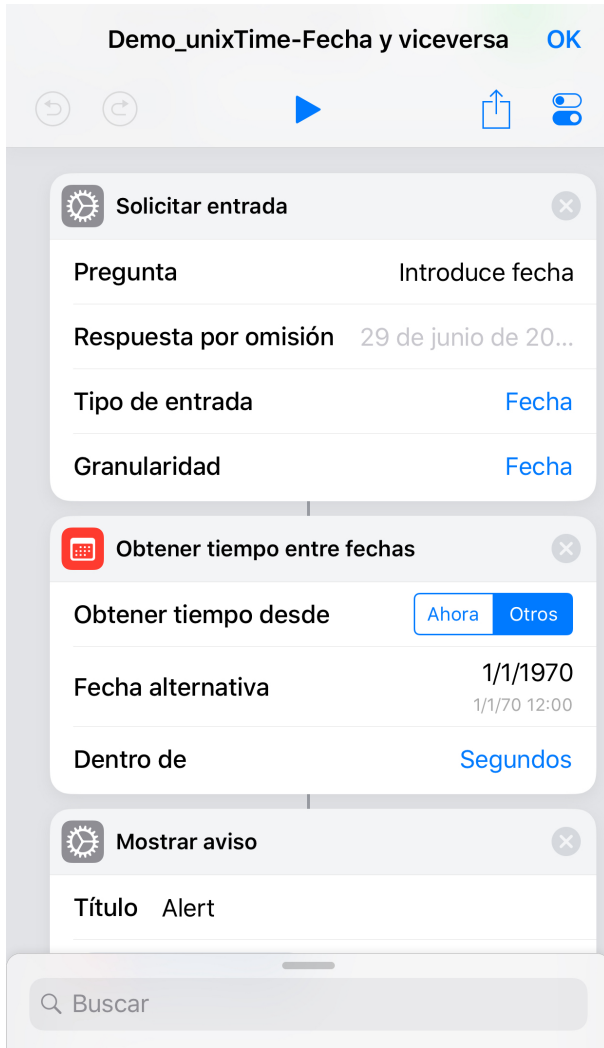
Observa este pequeño shortcut, sirve para poder convertir fechas a Unix Epoch y viceversa.



[DESCÁRGALO AQUÍ](#)



UNIX EPOCH > FECHA



FECHA > UNIX EPOCH

Este tipo de formato es muy utilizado por sistemas informáticos, por ejemplo, si eres usuario de [PLEX](#), debes saber que guarda los campos de fecha en este formato.

Como curiosidad, señalar que este punto en el tiempo no cambia técnicamente, sin importar dónde se encuentre en el mundo, es decir, no tiene en cuenta el desfase horario. Este hecho es una clara ventaja frente a formatos de fecha basados en UTC, pero por contra tiene la incomodidad del formato en si, que a simple vista es imposible saber a que fecha se está refiriendo.

El problema Y2K38

Como se ha dicho, numerosos sistemas informáticos usan esta marca de tiempo. El problema radica en que la mayoría de sistemas basados en arquitecturas de 32 bits definen el tipo de dato `time_t`, que es el usado para guardar este número de segundos, como un entero de 32 bits con signo. (representa números entre -2.147.483.648 y 2.147.483.647). Esto significa que el último segundo representable en este formato será a la 03:14:07 UTC del 19 de Enero de 2038. Llegados a este punto, se desbordará el contador y pasará a -2.147.483.648, indicando una fecha errónea de 1901, causando cálculos y procesamientos erróneos y convirtiéndose en un problema mundial.

Se ha dicho 32 bits, y como sabrás las nuevas tecnologías ya están en 64bits, por lo que este problema no sucedería al no producirse el desbordamiento. Concretamente, el problema se retrasaría 2,9E12 años, es decir, una solución permanente. Se espera que en 2038 no haya ya sistemas de 32 bits, por lo que este posible problema afectaría a sistemas realmente de forma muy puntual.

5.5. JSON

JSON (JavaScript Object Notation), es un formato de intercambio de datos abierto, como [XML](#) o [YAML](#), que, a diferencia de éste, usa una estructura de fácil lectura para humanos basado en una estructura de datos *clave:valor*.

A día de hoy, aunque JSON naciera dentro del lenguaje **Javascript**, se le considera un lenguaje de marcado propio, pues la mayoría de lenguajes de programación poseen herramientas para generar y leer datos en formato JSON.

Al decir que se basa en la estructura *clave:valor* se está indicando que el lenguaje usa un formato igual que los diccionarios de toda la vida, como el de la RAE (Real Academia de Española) en el que primero aparece la palabra a definir como en el siguiente ejemplo:

nombre: Decir el nombre de alguien o algo.

En este ejemplo **nombre** es la *clave* y **Decir el nombre de alguien o algo** es el *valor*. Esta es la base de JSON.

En el punto 6.3 de este libro estudiarás los **Diccionarios**, si, así se llaman, diccionarios, porque es lo que son, una clave y una definición del valor que se le asignan.

Las claves pueden tener el nombre que se quiera, y, como son cadenas de texto *Strings* deberán ir siempre entrecomilladas: "clave". Es muy importante tener en cuenta que **las claves deben ser únicas**, no puede haber dos claves con el mismo nombre.

Los valores que acepta *JSON* son:

- **Números:** números enteros, decimales o exponenciales, usando la notación E del modo 3.46E10 que es equivalente a 34600000000. Los números no necesitan entrecomillarse, por ejemplo: "nRuedas": 4 para definir el numero de ruedas de un coche.
- **Texto:** secuencias de letras o números que pueden estar vacíos o contener información. Deben ir entrecomillados, como se dijo antes y admite la barra invertida \como carácter de escape, es decir, para marcar aquellos caracteres que estén reservados en el lenguaje que se esté usando y evitar

ambigüedades. Por ejemplo "color":"rojo" para definir el color del coche.

- **Booleano**: es decir, de verdadero o falso. Las claves booleanas solo admite dos valores **true** para verdadero y **false** para falso y por tanto no es necesario entrecorillarlos. Por ejemplo: "arrancado": false para definir un estado del coche que solo admite verdadero o falso.
- **Lista**: una lista ordenada en los que cada elemento puede ser del tipo que sea (numero, texto, booleano, otra lista, otro diccionario o un valor nulo). Las listas se escriben con corchetes [] y los elementos se separan con comas.
- **Diccionarios**: si, eso es, uno de los valores que puede adoptar una clave, es precisamente otro diccionario, esto es, además, muy común como verás más adelante. Los diccionarios empiezan y acaban con llaves {}.
- **Valor vacío**: un valor vacío se representa bien con un texto sin contenido, es decir "" o bien con la palabra reservada null.

Los archivos JSON empiezan y acaban con llaves {} y los elementos se separan por comas , excepto el último elemento que no la necesita ya que tras él no hay ninguno más.

A continuación vas a ver un ejemplo sencillo de un archivo JSON para definir los atributos de un coche:

```
{
  "color": "negro",
  "nRuedas": 4,
  "nPuertas": 5,
  "encendido": false,
  "faros": ["posición", "cortas", "largas",
"antiniebla"]
}
```

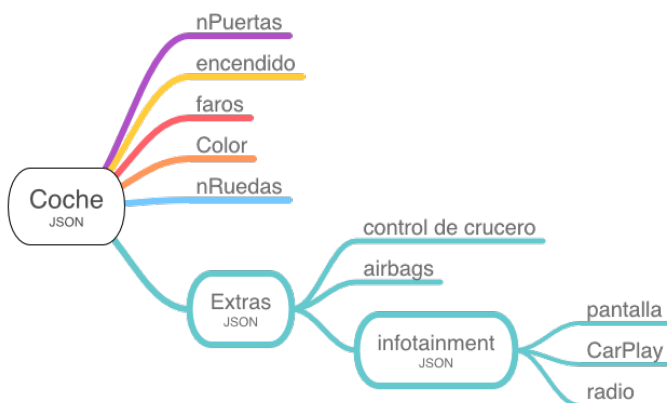
En este caso se han mezclado diferentes tipos de datos, el color es un *texto*, el número de ruedas un *número*, el número

de puertas es un *número*, el encendido es un *booleano*, y los faros son una *lista*.

Tal y como lo estas viendo puedes estar pensando aún que para qué necesitaría nadie introducir un diccionario dentro de otro. Lo más fácil, como siempre es verlo con un ejemplo, para ello, observa el mismo ejemplo del coche:

```
{
  "color": "negro",
  "nRuedas": 4,
  "nPuertas": 5,
  "encendido": false,
  "faros": ["posición", "cortas", "largas",
"antiniebla"],
  "extras": {
    "control de crucero": true,
    "airbags": ["conductor",
"acompañante", "laterales"],
    "infotainment": {
      "pantalla": 7,
      "CarPlay": true,
      "radio": ["am", "fm", "DAB"]
    }
  }
}
```

A tener en cuenta, los espacios entre líneas no afectan al formato, simplemente se acostumbran a hacer sangrías para que visualmente sea más sencillo de interpretar. Verás que aquí el esquema es este:



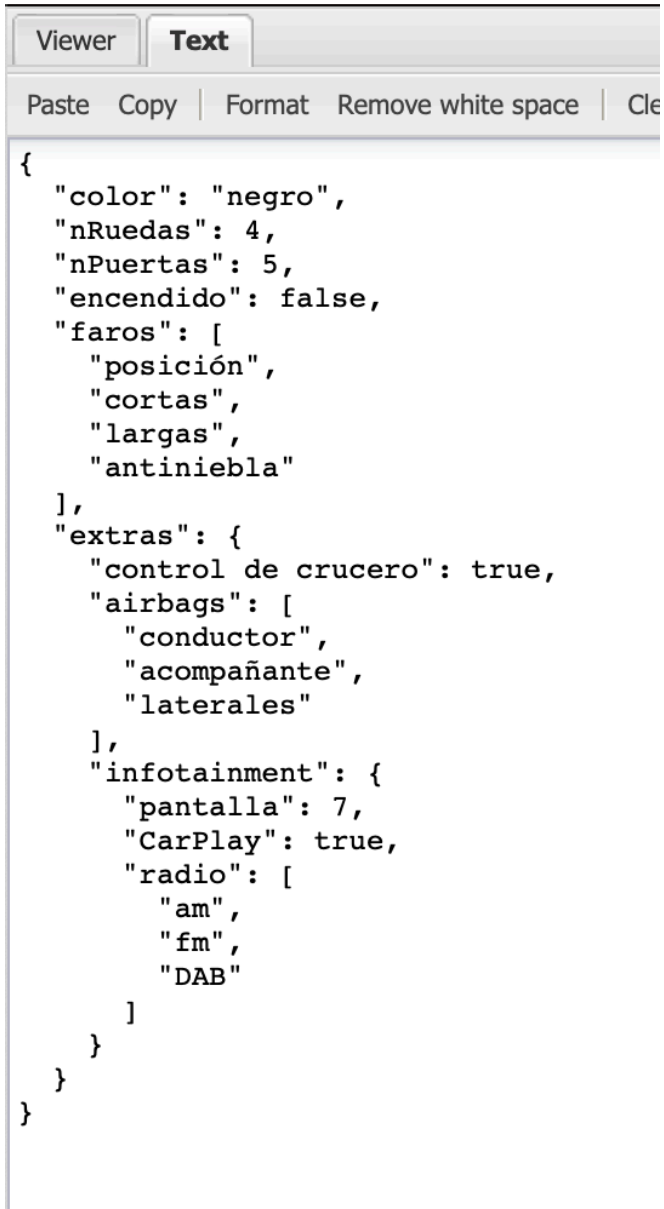
ESQUEMA DEL JSON

Es un diccionario JSON en el que se ha introducido otro diccionario en formato JSON dentro de la clave extras, y luego otro dentro de la clave infotainment.

Los archivos JSON son muy potentes y livianos como puedes ver. La forma de trabajar con ellos en Shortcuts se estudiará en el punto 6.3 del libro, ya que los diccionarios almacenan la información en formato JSON.

En archivos grandes puede ser un lío saber si el archivo JSON está bien construido. En ese caso, una buena herramienta para comprobar que la estructura es correcta es [JSONLint](#)

Otra gran herramienta es <http://jsonviewer.stack.hu/>. Con ella además de validar un texto en JSON podrás verlo en modo árbol, es decir, agrupando el contenido para facilitar su lectura y comprensión.



The image shows a screenshot of a web application titled "JSON Viewer" in "Text" mode. The interface includes a menu bar with options: "Paste", "Copy", "Format", "Remove white space", and "Clear". The main content area displays a JSON object representing a car's configuration. The JSON is formatted with indentation and line breaks for readability.

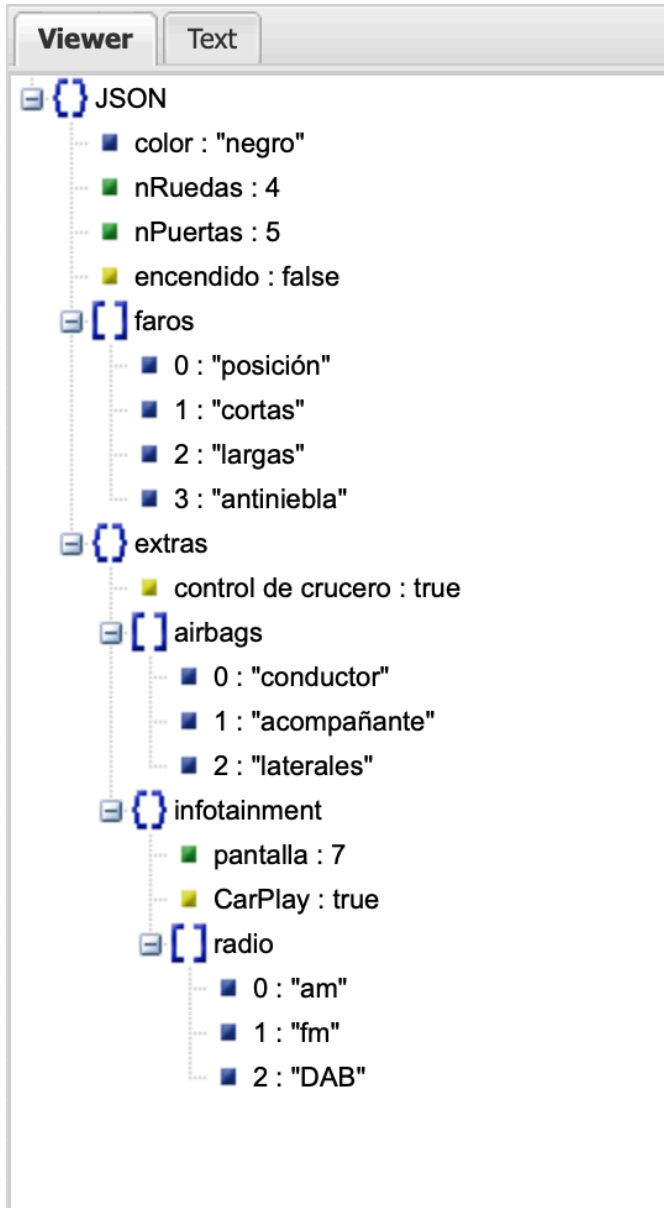
```
{
  "color": "negro",
  "nRuedas": 4,
  "nPuestas": 5,
  "encendido": false,
  "faros": [
    "posición",
    "cortas",
    "largas",
    "antiniebla"
  ],
  "extras": {
    "control de crucero": true,
    "airbags": [
      "conductor",
      "acompañante",
      "laterales"
    ],
    "infotainment": {
      "pantalla": 7,
      "CarPlay": true,
      "radio": [
        "am",
        "fm",
        "DAB"
      ]
    }
  }
}
```

JSONVIEWER EN MODO TEXTO

Viewer Text

JSON

- color : "negro"
- nRuedas : 4
- nPuertas : 5
- encendido : false
- faros
 - 0 : "posición"
 - 1 : "cortas"
 - 2 : "largas"
 - 3 : "antiniebla"
- extras
 - control de cruce : true
 - airbags
 - 0 : "conductor"
 - 1 : "acompañante"
 - 2 : "laterales"
 - infotainment
 - pantalla : 7
 - CarPlay : true
 - radio
 - 0 : "am"
 - 1 : "fm"
 - 2 : "DAB"



JSONVIEWER EN MODO ÁRBOL



Descárgalo aquí

Recuerda introducir el valor del token en el diccionario inicial, este parámetro se encuentra en las opciones de la App Bear, General -> Token de la API.

Este ejemplo contiene las siguientes opciones:

1. **Nueva Nota:** Permite al usuario dictar una nueva nota, o recuperar el texto del portapapeles para crearla. A destacar que el texto copiado puede ser texto enriquecido con enlaces, etc...Además se muestran los tags (etiquetas) existentes, permitiendo seleccionar una o más de ellas para clasificar la nota. Finalmente se pregunta al usuario si quiere incluir algún fichero, y si la nota debe ir anclada.(si debe estar priorizada en la lista de notas)
2. **Añadir a Nota:** Permite seleccionar un tag para filtrar las notas a buscar. El usuario dicta un texto y este se añade antes o después según se escoja.
3. **Añadir Fichero a Nota:** Como su propio nombre indica, adjunta un fichero a una nota seleccionada por el usuario.
4. **Buscar Nota:** Filtra notas por una o más palabras de búsqueda.
5. **Web -> Nota de Bear:** Solicita una URL al usuario y el contenido del enlace es convertido a markdown.

Como has podido comprobar, las capacidades de Bear desde Shortcuts son a día de hoy muy superiores a Notas, y ha servido para que puedas ver como una aplicación de IOS con

una buena implementación del protocolo x-callback-url permite jugar y mucho con Shortcuts.

5.5.1 JAYSON

Hasta ahora has podido ver cómo se estructura un fichero JSON y en el apartado anterior se han definido 2 herramientas online para poder inspeccionar y verificar si un determinado fichero tienes errores o simplemente, se desea poder leerlo de un modo más visual.

Bien, para iOS tienes disponible una aplicación llamada JAYSON. Esta app es gratuita con una única compra in app y sin duda es muy útil para nuestros propósitos a la hora de depurar shortcuts. El autor es Simon Støvring, desarrollador indie, y está especializada en leer ficheros json.

El modo más intuitivo de usarla es, sencillamente, abriendo la app y accediendo a cualquier JSON que se desee usando el explorador de ficheros nativo de iOS para localizarlo. También instala acciones en la app Shortcuts. Esta segunda opción tiene "truco", y es que debido a la imposibilidad de que JAYSON reciba el fichero de Shortcuts, utiliza un pequeño rodeo que han usado otras apps para poder "intercambiar" información, el portapapeles.

Antes de ver como hacer todo esto, véanse las acciones disponibles de JAYSON en Shortcuts:

Las cuatro acciones son básicamente la misma forma de visualizar el JSON que reciba. Todas se basan en lo mismo, antes de la acción de JSON (sea la que sea), debes poner en el portapapeles el JSON que recibirá JAYSON.

Un ejemplo es la mejor forma de verlo, fíjate en estas acciones:

Sugerencias de Siri



View JSON in App

Opens document in the app.



View JSON in Notification

Suitable for quickly previewing JSON in the Short...



View JSON in App and Save File

Opens document in the app. Saves document on...



View JSON in App and Save File

Opens document in the app. Saves document in i...

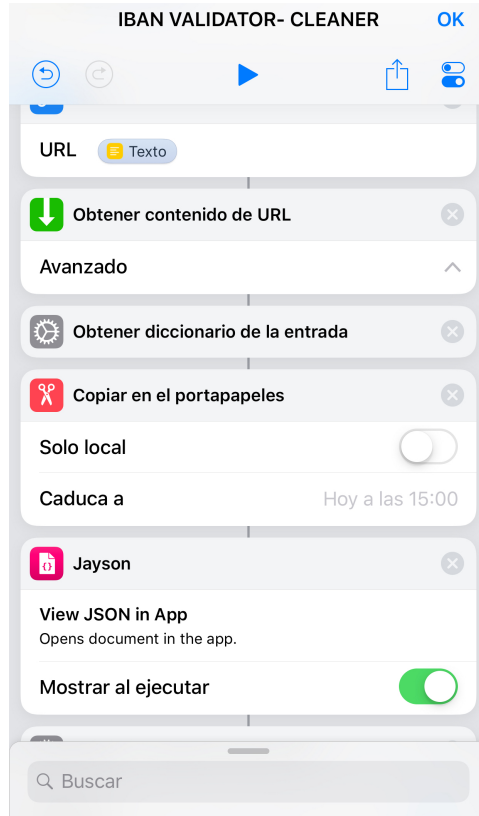


Se “lanza” una URL que espera devolver una estructura JSON y antes de utilizar la acción de la aplicación para poder visualizarlo, se coloca una acción “Copiar al Portapapeles”. Sólo de este modo se consigue que JAYSON reciba el fichero y puedas ver algo como esto:

La aplicación no tiene ningún misterio, pero es extremadamente sencilla y útil. Hace lo que debe hacer, que es :

- Mostrar en detalle cada nodo del JSON
 - Permite abrir pestañas.
- * Hacer una división vertical para poder visualizar un mismo JSON en puntos distintos.

- * Visualizar el fichero en modo texto o en modo árbol, indicando en la parte superior la ruta en la que se está en cada momento.



En definitiva, si vas a tratar con JSONs grandes es más que recomendable usar una herramienta de este tipo para ahorrar dolores de cabeza y tiempo al navegar por los datos de forma estructurada. Además es muy fácil detectar errores de formato típicos al construir tus propios diccionarios JSON.

< Files

validation 2 splits +

jayson.json > result > validation

```
{
  "account_checksum": "Valid",
  "iban_checksum": "Valid",
  "iban_length": "Valid",
  "iban_structure": "Valid",
  "iban_validity": "Valid"
}
```

< Files

2 splits +

jayson.json > result > validation

- account_checksum String Valid >
- iban_checksum String Valid >
- iban_length String Valid >
- iban_structure String Valid >
- iban_validity String Valid >

ult > validation > iban_checksum

Valid

5.6. Base64

En Shortcuts no es imprescindible haber programado para poder alcanzar la excelencia, pero si lo has hecho es probable que alguna vez hayas oído hablar de [Base64](#) para incluir ficheros en un JSON por ejemplo, y también es probable que hayas tenido algún que otro problema con caracteres de escape una vez codificados... Eso es harina de otro costal, pero en este apartado se intentará explicar que es una codificación Base64 y que usos puede tener para nuestros atajos.

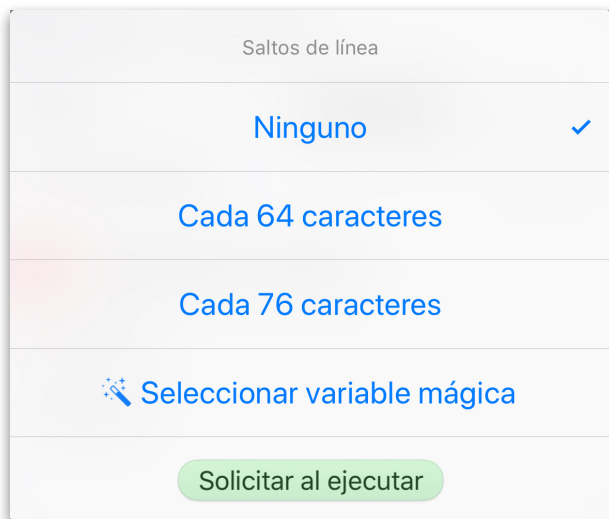
Antes de empezar con la explicación es necesario introducir dos conceptos importantes y que son parte de la cultura del programador:

- **archivo binario:** es un archivo que contiene cualquier tipo de información codificada en código binario (0 y 1). Un ejemplo clásico son las fotografías, la música, películas, pdfs o los ejecutables de los programas. Si abres un archivo binario con un programa que lea textos, como TextEdit en mac o NotePad en windows verás que es una cadena ininteligible de letras y símbolos.
- **archivo de texto plano:** es un archivo cuyo contenido es únicamente texto, por tanto es necesario interpretarlos para leerlos. Cualquier programa que abra archivos de texto podrá abrir el archivo y un humano podrá entenderlo.

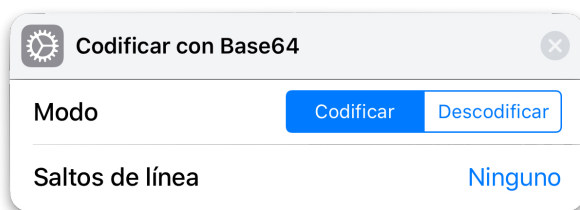
¿Qué ocurre si en un shortcut quieres incluir una imagen sin que el usuario tenga que guardarla en su carrito, en Dropbox o iCloud? Ahí es donde Base64 viene a rescate ya que es un grupo de esquemas de codificación que es capaz de transformar/codificar un fichero binario a texto. Es decir, por ejemplo dispones de una imagen, y al codificarla con Base64 obtendrás un conjunto de caracteres y números que, haciendo el proceso inverso, son capaces de recuperar la imagen original.

En muchas API que requieren el envío de ficheros verás como previamente hay que codificarlos para poder realizar la petición, por ejemplo en la API de [cloudConvert] (<https://cloudconvert.com/api/conversions>).

Fíjate en la captura del bloque:

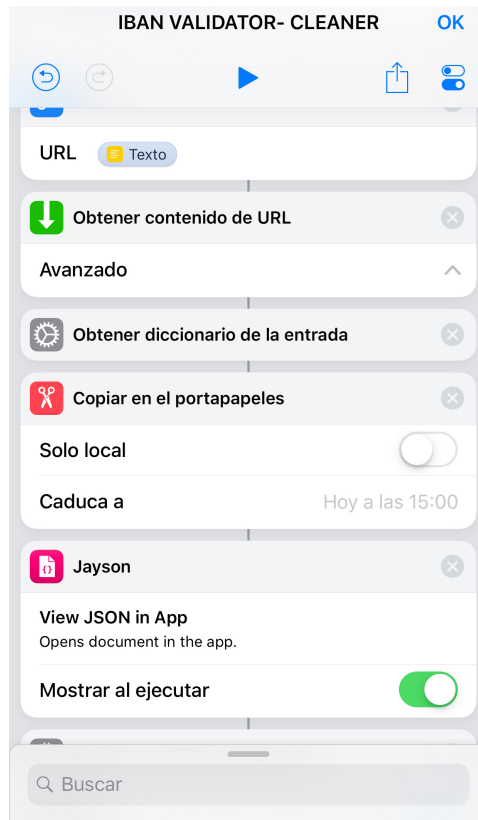


BASE64



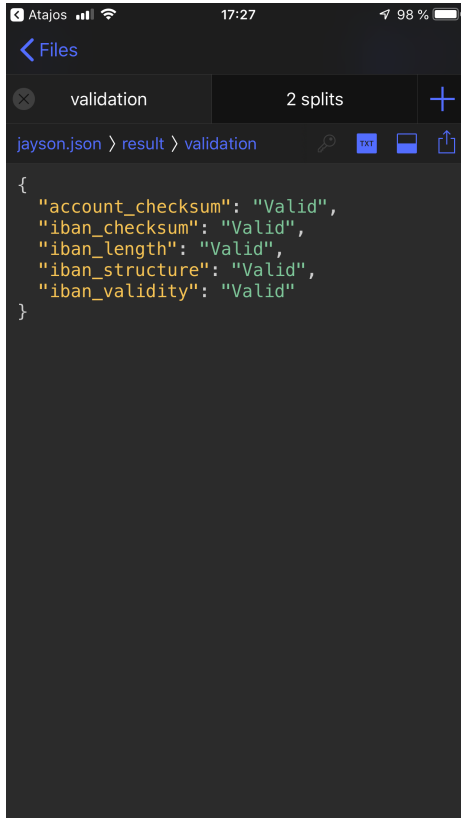
OPCIONES CODIFICACIÓN

MIME (Multipurpose Internet Mail Extensions) no especifica un tamaño fijo para las líneas codificadas en base64, pero sí precisa un tamaño máximo de 76 caracteres. Muchas implementaciones usan los caracteres CR/LF (retorno de



carro y salto de línea) para delimitar las líneas codificadas. De esta manera, el tamaño real de los datos codificados conforme a MIME suele ser de un 140% del tamaño original.

Otro uso que es habitual para este tipo de bloque es decodificar la información de un bloque de texto previo donde hay un fichero codificado en Base64. El resultado de todo esto es que un usuario puede insertar un PDF explicativo por ejemplo, una imagen, etc.... En el ejemplo siguiente se aprecia como en un bloque de "Texto" se pega el fichero codificado, y en el siguiente se descodifica de tal modo que el usuario cuando ejecute el atajo verá el PDF.



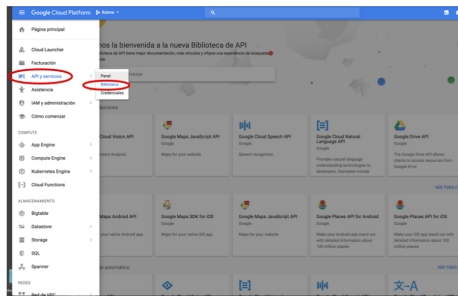


DESCODIFICAR BASE64

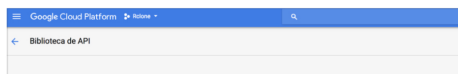


Google Console Crear Proyecto - API

1. Tener una cuenta de Gmail y estar dispuesto a dar los datos de una tarjeta de crédito a Google (no cobrarán NADA sin tu consentimiento), y te regalan 300\$ el primer año (no los gastarás, tranquilo !!!)
2. [Google Cloud Platform](#)
3. Logarte con user/pass
4. Vamos a API ► Biblioteca



5. Seleccionamos Google Cloud Translation API y la habilitamos.



PDF RESULTANTE

6. Trabajando con archivos

Hay dos maneras de trabajar con archivos en Shortcuts, unos son los archivos procedentes del carrete de fotografías, todos ellos son **archivos multimedia**, es decir, fotografías o vídeos. Y el otro es mediante la integración con **“Archivos”**, la app nativa para conectar todas tus nubes en un solo lugar, y Dropbox. Estos archivos serán de cualquier tipo.

Verás que es muy difícil separar el trabajo con archivos multimedia y con archivos de cualquier tipo, por lo que verás referencias a unos y a otros en cada uno de los ejemplos explicados.

6.1 Trabajando con archivos multimedia

Todas las acciones que ofrece Shortcuts para trabajar con archivos del carrete o recuperados desde Archivos o Dropbox pero que sean de carácter multimedia: fotografía, vídeo o audio se encuentran agrupadas bajo el epígrafe ****Fotos y vídeo"** de la lista de acciones de Shortcuts.

Para que veas como funcionan algunos de ellos se van a hacer un par de ejemplos.

6.1.1 Firmar fotografías

En este ejemplo se va a usar shortcuts para aplicar una marca de agua o una firma a tus fotografías para que al compartirlas todo el mundo sepa que son tuyas. Hay apps de pago, y bastante caras que hacen esto mismo que con Shortcuts se resuelve de manera sencilla.

Este shortcut será del tipo extensión, por lo que al estar en el carrete, enviarás una fotografía a Shortcuts para que te la devuelva firmada.

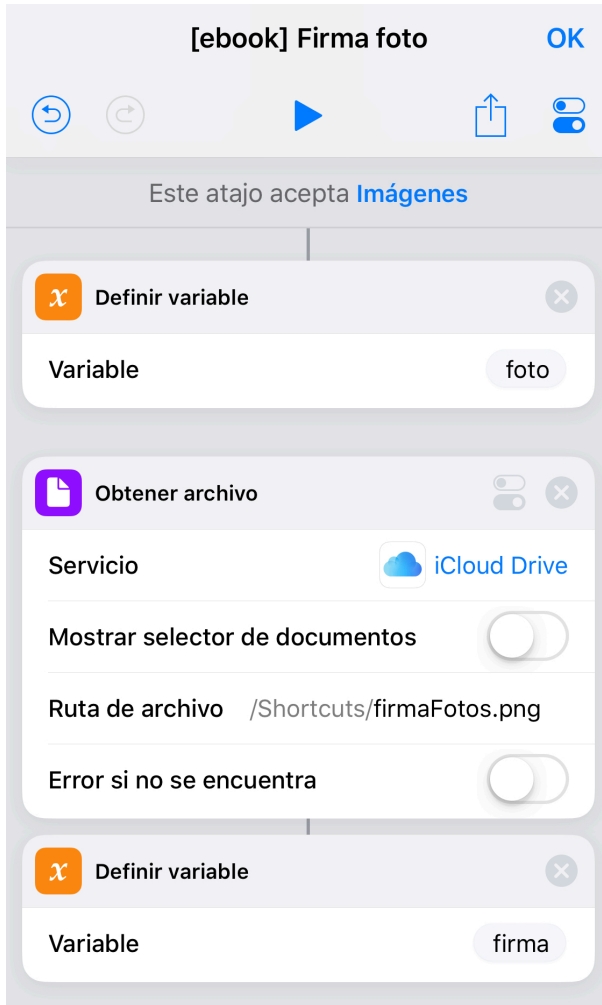
Para ello en primer lugar deberás tener un archivo png con la firma o la imagen que quieras usar de marca de agua en la carpeta */shortcuts/* de tu cuenta de iCloud.

Guarda en una variable la imagen que es la entrada del shortcut pues se va a necesitar recuperar varias veces para obtener sus atributos.

A continuación recupera también la firma desde tus archivos, para ello usa la acción “Obtener archivos” de esta manera:

- Servicio: iCloud Drive (podría ser Dropbox)
- Mostrar selector de documentos: desactivado
- Ruta del archivo: */shortcuts/firmaFotos.png*
- Error si no se encuentra: desactivado

Y se guardará en la variable “firma”.



COMIENZO DE LA CONFIGURACIÓN DEL SHORTCUT DE FIRMAS

Ahora tienes en la variable "foto" la fotografía que vas a firmar y en la variable "firma" la firma que vas a usar, por lo que está todo preparado para empezar.

Para ello recupera la variable “foto” y mediante la acción “Obtener detalles de archivos” recupera el nombre que tiene la fotografía:

- Obtener: Nombre

Recupera de nuevo la variable “foto” y esta vez obtén la anchura de la imagen. Para ello usa la acción “Obtener detalles de imágenes”:

- Obtener: anchura

De la misma manera, vuelve a recuperar la variable “foto” y obtén la altura de la misma.

- Obtener: altura

Se necesita del ancho y el alto para saber si la imagen es vertical u horizontal. Mediante un “Si” a continuación de la altura, configurado así:

- Entrada: Es mayor que
- Número: variable mágica “anchura”

Se puede saber la posición, ya que si la altura es mayor que la anchura será una imagen vertical. Y si la anchura es mayor que la altura la imagen será horizontal.



MÉTODO PARA SABER SI LA IMAGEN ES VERTICAL U
HORIZONTAL

En el apartado “Si”, que es cuando se cumple que la imagen es vertical, recupera el valor de la altura mediante “Obtener variable” y seleccionado la variable mágica donde se recuperaba la altura de la imagen. Esta altura se divide por 8, y así obtendremos los pixeles de altura de la imagen de firma. Es decir $1/8$ de la altura de la fotografía cuando es vertical.

Del mismo modo, cuando es horizontal, en el apartado “si no” obtén la misma altura y divídela entre 6, ya que al ser horizontal la altura deberá ser mayor para que quede igualmente legible.

El 8 y el 6 de la división son valores que han funcionado bien en el ejemplo, pero que puedes cambiar para adaptarlo más a tus gustos.

El resultado del bloque “Si” será una altura de firma que tendrá en cuenta la posición de la fotografía, por lo que puedes proceder al firmado, para ello recupera la variable “foto” de nuevo y usa la acción “Superponer imagen” con esta configuración para aplicar la firma:

- Imagen: variable “firma”
- Mostrar el editor de imagenes: desactivado
- Posición: abajo derecha
- Anchura: automática, de esta manera te aseguras no deformar la firma
- Altura: variable mágica que recoge el resultado del “Si”, que era la altura de la firma con respecto a la imagen seleccionada
- Rotación (grados): 0
- Opacidad: 100



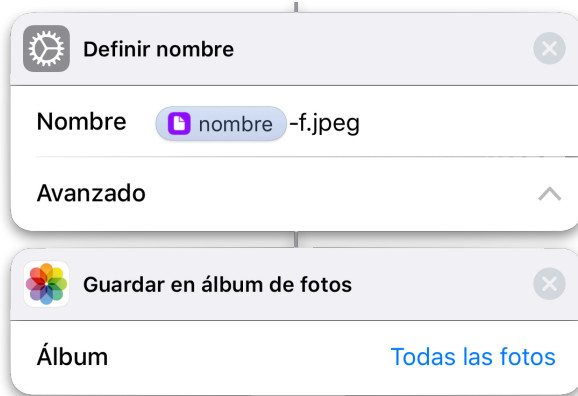
CONFIGURACIÓN DE LA FIRMA

Ahora solo queda definir el nombre de la nueva imagen mediante la acción “Definir nombre”:

- Nombre: recupera la variable mágica donde se obtenía al principio del todo el nombre del archivo y añade un -f para diferenciar que está firmada y añade la extensión que será.png. Es decir:
{nombre}-f.png
- El campo avanzado déjalo sin tocar.

Y ahora mediante una acción “Guardar en álbum de fotos” vuelve a guardar la imagen en el carrete:

- Álbum: todas las fotos.



CONFIGURACIÓN DEL GUARDADO



[DESCÁRGALO AQUÍ](#)

6.1.2 Reducir el tamaño de una fotografía

De nuevo se propone un ejemplo sencillo para suplir una necesidad que, a menudo, viene en forma de app de pago en la AppStore y que puedes hacer en pocos pasos con Shortcuts.

Este ejemplo se configurará para que funcione como extensión, ya que es más sencillo ver las fotografías en la app de Fotos y mandarlas a Shortcuts.

Se hará de forma que se pueda elegir cuánta reducción se va a aplicar: 50% ó 25%.

Comienza un nuevo shortcut y guarda en una variable la imagen de la entrada. Recuerda, que en la configuración del shortcut deberás seleccionar la opción “Mostrar al compartir” y

que en los tipos soportados es mejor seleccionar solo imágenes.

15:30 ↖



Ajustes

OK

Nombre

Redimensionar imagen >

Icono



Añadir a Siri

Añade este atajo a Siri para ejecutarlo con tu voz.

Mostrar en el widget



Mostrar al compartir



Tipos aceptados

Imágenes >

Usa este atajo desde otras apps. Para ello, abre la hoja para compartir y pulsa Atajos.

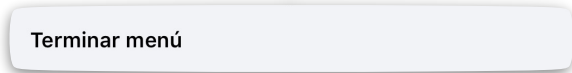
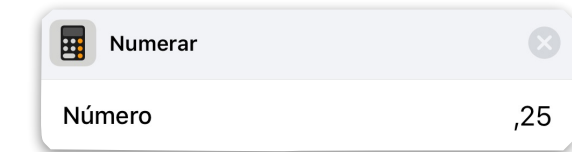
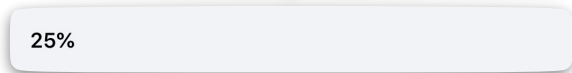
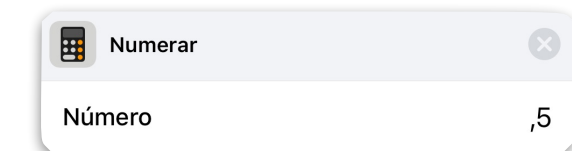
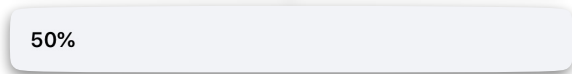
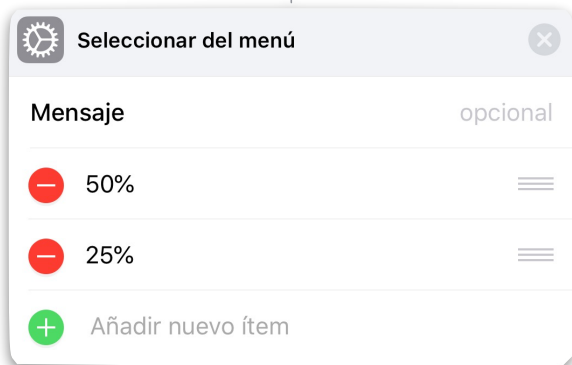
Añadir a pantalla de inicio

Compartir atajo

Preguntas de importación >

CONFIGURACIÓN DEL SHORTCUT COMO EXTENSIÓN PARA IMÁGENES

Ahora añade una acción “Seleccionar del Menú” y añade dos opciones: 50% y 25%. Dentro de cada una de las opciones añade un “numerar”, en el de 50% escribe 0,5 y en el de 25% escribe 0,25.



CONFIGURACIÓN DEL MENÚ PARA REDUCIR

Como el shortcut admitirá que se envíen varias imágenes, a continuación recupera la variable “Foto” y usa un “Repetir con cada uno”.

Ahora, dentro del bloque repetir, mediante “Obtener detalles de imágenes” obtén uno de los valores de ancho o alto, da igual. Para el ejemplo se hará con la altura.

Usa un bloque calcular y selecciona la operación multiplicar. En el valor usa la variable mágica de la salida del menú en el que se seleccionaba la calidad.

Mediante “Obtener variable” seleccionando la variable especial “repetir ítem” recupera la fotografía que se está procesando en este momento y arrastra una acción “Cambiar tamaño de imagen” a continuación:

- Anchura: dejar en blanco.
- Altura: selecciona la variable mágica del resultado de la operación de multiplicación.



CONFIGURACIÓN DE BLOQUE "REPETIR CON CADA UNO"

Termina con la acción "Guardar en álbum de fotos:

- Album: Todas las fotos

Y con esto se guardará una nueva imagen que es idéntica a la anterior pero con la reducción que hayas seleccionado de tamaño.



[DESCÁRGALO AQUÍ](#)

6.1.3 Convertir un vídeo

A menudo se graba con nuestros dispositivos en calidad 4k o 1080p, pero dependiendo de cómo quieras compartirlo puede ser que el tamaño del archivo sea demasiado grande.

En este ejemplo vas a aprender a cambiar el tamaño de un vídeo para que ocupe menos, a costa de reducir su calidad.

Este shortcut será “independiente”, es decir, el usuario deberá seleccionar el vídeo desde el propio shortcut. Para ello añade una acción del tipo “Seleccionar fotos” y a continuación la acción “Codificar contenido” con esta configuración:

- Solo audio: desactivado, si lo activaras, serviría para obtener el sonido y eliminar el vídeo, por ejemplo para obtener una canción de un vídeo musical.
- Tamaño: Selecciona 1280x720p. De esta manera seguirá siendo HD pero de menor calidad y mucho menor tamaño.
- Velocidad: normal. Verás que si abres el menú desde aquí podrías acelerar el vídeo o ralentizarlo.
- En metadatos deja todo en blanco, pero lee las opciones para saber hasta donde puedes llegar con esta acción tan completa.

Ahora añade una acción "Guardar en álbum de fotos:

- Album: Todas las fotos

Y con esto se guardará un nuevo vídeo con el mismo contenido pero reducido a 720p, por lo que será apto para compartir sin gastar la conexión de datos.



[DESCÁRGALO AQUÍ](#)

6.1.4 Crear un gif a partir de un vídeo

Ahora que están de moda los gifs de nuevo, puedes usar Shortcuts para convertir un vídeo o un trozo de un vídeo en un gif que puedas compartir donde más te apetezca.

Para ello comienza un nuevo shortcut y añade una acción del tipo “Seleccionar fotos”, con seleccionar varias desactivado.

Ahora usa la acción “Acortar contenido” para que se presente una interfaz donde seleccionar un trozo corto del vídeo, ya que los gifs suelen ser así.

Añade una acción “Crear Gif”:

- Segundos por foto: 0,1
- Blucle infinito: activado
- Tamaño automático: activado

Ahora añade una acción "Guardar en álbum de fotos:

- Album: Todas las fotos

Recuerda que los gifs son imágenes en movimiento y por tanto no tienen audio.



[DESCÁRGALO AQUÍ](#)

6.2 Trabajando con cualquier tipo de archivos

Toda la capacidad de trabajar con archivos desde Shortcuts se encuentra bajo el epígrafe “Documentos” y las descripciones son bastante útiles por si mismas. Es por ello

que se van a hacer algunos ejemplos útiles para entender la lógica de funcionamiento.

Ya en el apartado anterior se ha visto cómo recuperar el archivo para la firma de las imágenes, ahora se ahondará un poco más.

6.2.1. Unir varios PDF en uno solo

Ocurre frecuentemente, te mandan varios archivos PDF que querrías unir en uno solo, de nuevo con shortcuts y sin gastar nada en ninguna app externa lo puedes hacer muy fácil para ello se va a iniciar un nuevo shortcut en modo “independiente” para ver como hacer la selección de los archivos.

Usa una acción “Obtener archivo”:

- Seleccionar servicio: iCloud Drive o Dropbox, el que uses.
- Mostrar selector de documentos: activado
- Seleccionar varios: activado

Ahora usa la acción “Crear PDF”:

- Incluir margen: desactivado
- Incluir: todas las paginas

Con la acción “Guardar archivo” podrás volver a guardar todos los PDFs en un único archivo:

- Servicio: iCloud Drive o Dropbox, el que uses.
- Preguntar donde guardar: activado

Podrías sustituir la acción “Guardar archivo” por la de “Compartir”, en este caso te aparecería el menú compartir para enviar el PDF a la aplicación Archivos o Dropbox, cuyo efecto es el mismo que el anterior, o incluso adjuntarlo a un mail o enviarlo por WhatsApp o Telegram.



[DESCÁRGALO AQUÍ](#)

6.2.2. Página web a PDF

Hay veces en que te interesa guardar el contenido de una web en un PDF, por ejemplo, para guardar el comprobante de un pago en una compra online. De nuevo, con Shortcuts y muy pocos casos puedes conseguir resolverlo.

Para ello en un shortcut nuevo configurado para que sea del tipo “compartir” haz que solo pueda recibir URLs.

16:27



Ajustes

OK

Nombre

Web a PDF >

Icono



Añadir a Siri

Añade este atajo a Siri para ejecutarlo con tu voz.

Mostrar en el widget



Mostrar al compartir



Tipos aceptados Páginas web de Safari, Dire... >

Usa este atajo desde otras apps. Para ello, abre la hoja para compartir y pulsa Atajos.

Añadir a pantalla de inicio

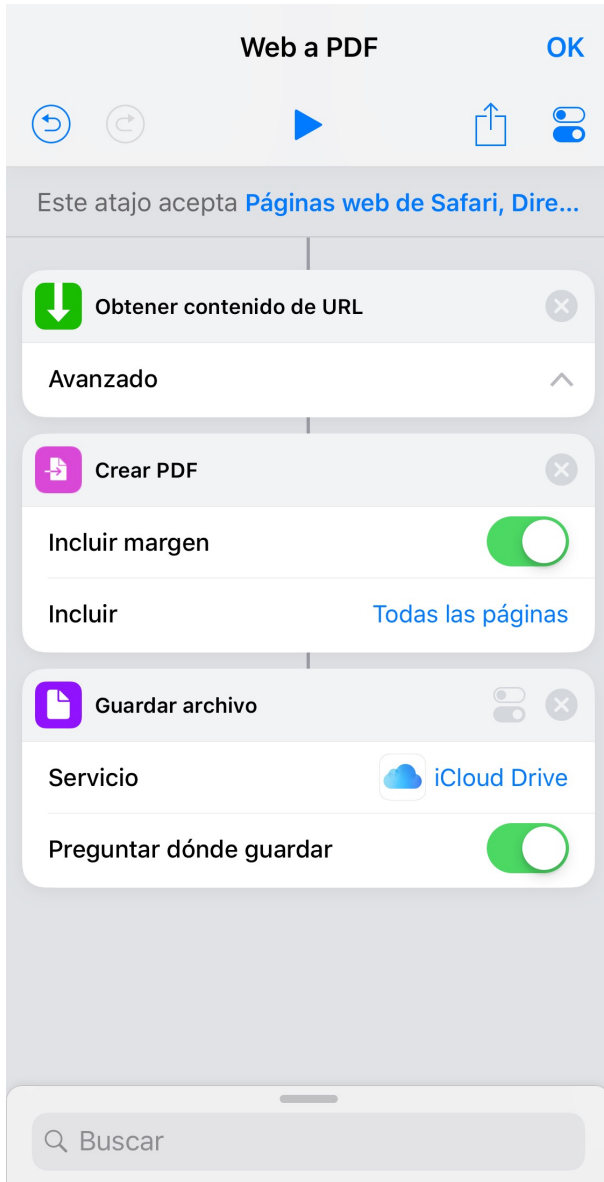
Compartir atajo

Preguntas de importación >

CONFIGURACIÓN DEL SHORTCUT PARA ADMITIR WEBS

Añade una acción “Obtener contenido de URL” y a continuación otra de “Crear PDF”. En la de crear PDF activa la opción de incluir margen.

Luego añade la acción “Guardar archivo” en el servicio que prefieras: iCloud Drive o Dropbox; y Preguntar dónde guardar activado.



CAPTURA DE TODO EL SHORTCUT

Estos pasos son suficientes para obtener un PDF a partir de una web que estes viendo en Safari.

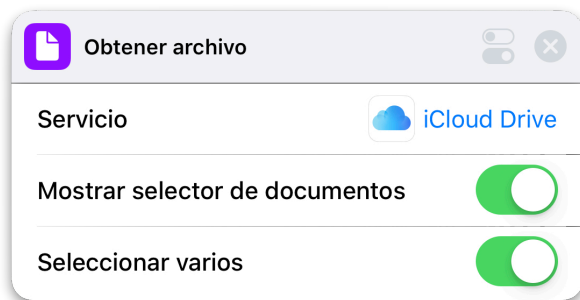


[DESCÁRGALO AQUÍ](#)

6.2.3. Comprimir varios archivos y compartirlos usando Dropbox

Este es otro caso típico. Tienes varios archivos, ya sean fotografías o documentos que necesitas enviarle a alguien mediante Dropbox.

Configura un nuevo shortcut, esta vez va a ser uno independiente. Añade, en primer lugar la acción “Obtener archivo” con el servicio de nube que uses y el selector de documentos y la opción para seleccionar varios archivos activado.



ACCIÓN PARA SELECCIONAR UN DOCUMENTO

Ahora usa una acción “Crear archivo comprimido” configurada así:

- Nombre del archivo comprimido: variable {fecha actual}-compartir:

- Formato de fecha: corto
- Formato de hora: ninguno
- Formato: zip, verás que si pulsas sobre ZIP te permite seleccionar más formatos de los que seguramente conozcas.

Ahora añade la acción “Guardar archivo” y esta vez selecciona como servicio “Dropbox” con preguntar dónde guardar activado.

Busca la acción “Obtener enlace a archivo” y añádela y a continuación la acción “Compartir”.

De esta manera el archivo se guardará en Dropbox, éste generará un enlace de archivo compartido que podrás enviar por el medio que quieras.



[DESCÁRGALO AQUÍ](#)

6.2.4. Editar Fotos o PDF.

Usando la acción de “Marcar”, también en “Fotos y vídeo” se pueden editar de forma rápida (equivalente a la edición nativa de IOS) PDF o fotografías.

Mira este [ejemplo](#) Con unos pocos bloques se ha fabricado un shortcut capaz de editar fotos, con el añadido que a partir de aquí podrías, por ejemplo, automatizar la creación de un email, WhatsApp, o lo que sea, para automatizar alguna tarea que hagas constantemente.

Bloque Marcar(editar)

OK



Seleccionar del menú



Mensaje

opcional



Editar Foto



Editar PDF



Añadir nuevo ítem

Editar Foto



Seleccionar fotos



Seleccionar varias



Editar PDF



Obtener archivo



Servicio



iCloud Drive

Mostrar selector de documentos



Seleccionar varios



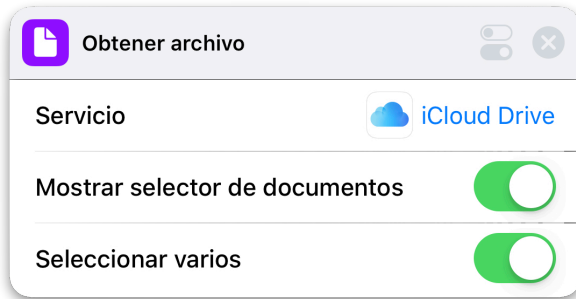
Terminar menú



Marcar



Un último aspecto a tener en cuenta, una vez acabada la edición, si pulsas en el icono del lápiz en la parte superior derecha, cambia la parte inferior y sale de edición, entrando en el modo compartir, para poder guardar, enviar la captura a otra app.



COMPARTIR

7. Conceptos avanzados

Hasta ahora, a lo largo de los capítulos precedentes, has tenido una toma de contacto con Shortcuts, y su forma de trabajo. Ya vas entendiendo la lógica de como crear un shortcut.

Además, se ha hecho una introducción de conocimientos colaterales a toda tarea de programación, como son RegEx, Markdown, trabajar con fechas, JSON o la codificación/descodificación en BASE64.

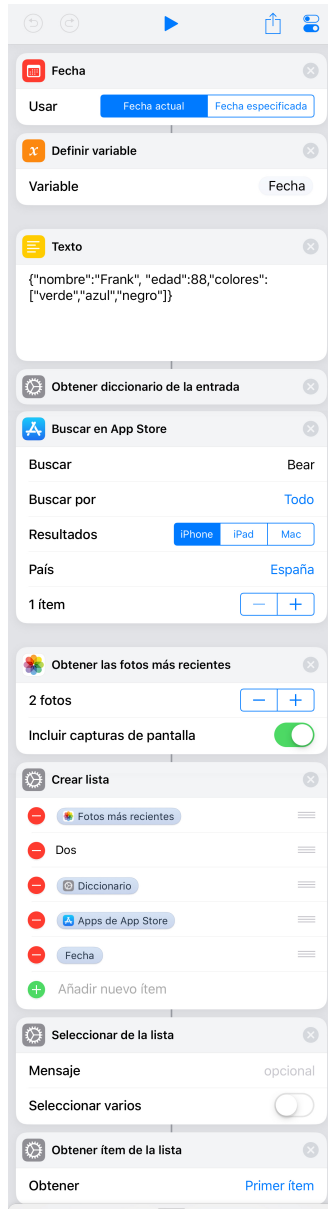
En este capítulo abordaremos el uso de estas tecnologías en Shortcuts para empezar a hacer shortcuts mas elaborados que sean capaces de dar respuesta a problemas más complejos, lo que abrirá mucho el abanico de posibilidades de esta app.

De nuevo, algunos temas, como reutilización de código, diccionario o expresiones lógicas tienen un uso más allá de la propia Shortcuts, aunque aquí se estudiarán de forma aplicada.

7.1 Listas

Las listas son un tipo de objeto que permite guardar cualquier “cosa” de una forma ordenada. En este capítulo se explicará como usarlas y, lo más importante, como explotaras al máximo para poder tener un recurso más para tus shortcuts.

Se puede definir una lista como un repositorio donde guardar cualquier tipo de objeto. Como muestra, en el ejemplo siguiente se almacenan distintos tipos de variable y se muestran en la acción *“Seleccionar de la lista”*.



El resultado sería este:



Las tres acciones que actúan directamente sobre una lista son: “Crear Lista”, “Obtener Ítem de la lista”, y la mencionada “Seleccionar de la Lista”. Los dos primeros bloques son de fácil comprensión:

“*Crear Lista*”: Permite añadir elementos a la lista, como se ha podido ver, cualquier elemento, variables, texto, objetos “App Store”, diccionarios...

“*Seleccionar de la Lista*”: Permite al usuario seleccionar un elemento de la lista y pasarlo al bloque siguiente. Esta selección es, pues, manual, depende del usuario. Además presenta una opción llamada “*Seleccionar Varios*” que permite seleccionar varios elementos que formarán una nueva lista a la salida del bloque.

“*Obtener Ítem de la Lista*”: Este bloque permite, de forma “programática”, es decir, sin intervención del usuario, recuperar un objeto de la lista. Las listas se enumeran de 1 a n elementos (lo habitual en los lenguajes de programación es que el primer elemento sea el 0).

Si lo deseas, instala el ejemplo de la captura anterior:



[DESCÁRGALO AQUÍ](#)

Es fácil ver como al crear la lista se añaden los distintos tipos de objeto, y al ejecutarlo, se observa como la lista creada va más allá que simples ítems de texto. En el siguiente apartado se explicará más en detalle el uso de las Listas Enriquecidas.

7.1.1. Listas enriquecidas

Desde la versión 1.6 de Workflow (recuerda, la antecesora de Shortcuts), publicada en Noviembre de 2016, se dotó al bloque “Seleccionar de la Lista” de una característica nueva: La capacidad de mostrar cada elemento de esa lista de una manera más rica y completa que hasta ese momento.



Buscar en App Store



Buscar

Plex

Buscar por

Todo

Resultados

iPhone

iPad

Mac

País

España

5 ítems



Repetir con cada uno



Definir nombre



Nombre

Repetir ítem

Avanzado



Terminar repetición



Seleccionar de la lista



Mensaje

opcional

Seleccionar varios



Q Buscar



[DESCÁRGALO AQUÍ](#)

Es muy importante entender como Shortcuts es capaz de obtener de cada “objeto” los distintos tipos que éste contiene. En el ejemplo anterior se aprecia claramente como se realiza una búsqueda a través del bloque “*Buscar en App Store*” y lo que devuelve es el nombre de la app, el desarrollador, el precio y el logo de la misma. Es decir que a pesar de todos estos tipos distintos, el motor de la aplicación es capaz de simplificar entre bambalinas todo el proceso que sería presentar todos los datos en una lista y lo ejecuta de forma transparente para el usuario.

No hace falta decir que, bloques de diferentes temáticas (fotos, AppStore, iTunes Store) que a su vez tienen acciones del tipo “*Obtener Detalles de ...*” son buenos candidatos para pasarlos a una acción del tipo “*Crear Lista*” y obtener de forma automática listas enriquecidas.

En los siguientes enlaces puedes ver algunos ejemplos:

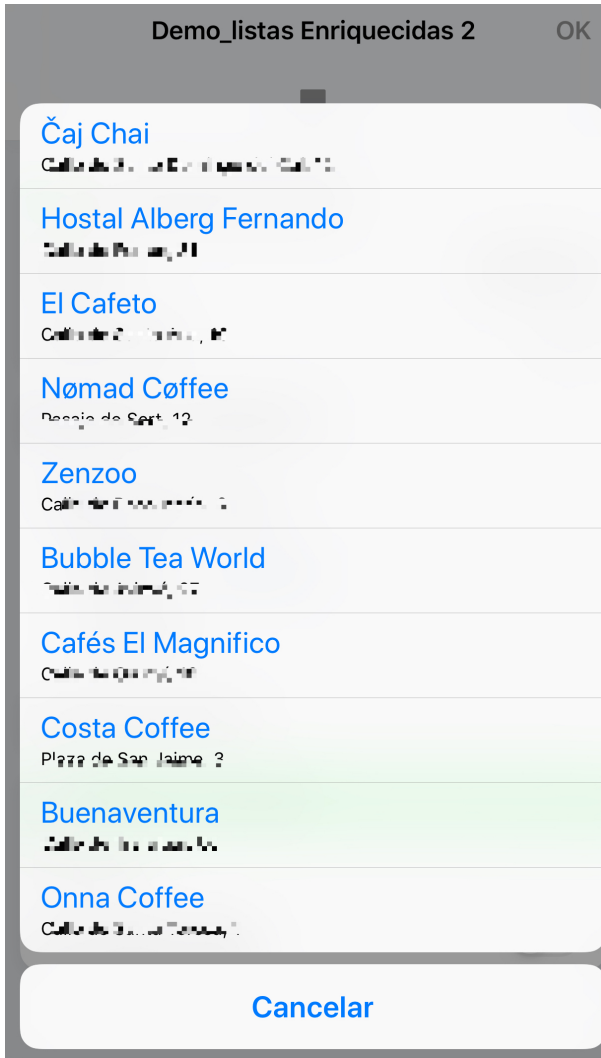
1. “*Buscar Negocios locales*”.



[DESCÁRGALO AQUÍ](#)



Con este resultado:



2. "Buscar en iTunes Store".



[DESCÁRGALO AQUÍ](#)



Buscar en iTunes Store



Buscar

Michael jackson

Categoría

Música

Buscar por

Álbum

Resultados

Álbumes

País

Estados Unidos

10 ítems



Repetir con cada uno



Definir nombre



Nombre

Repetir ítem

Avanzado



Terminar repetición



Seleccionar de la lista



Mensaje

opcional











Seleccionar varios



🔍 Buscar

Con este resultado:

Demo_listas Enriquecidas 3 OK

	The Essential Michael Jackson... 16,99 US\$ Michael Jackson
	Michael Jackson's This Is It (Th... 10,99 US\$ Michael Jackson
	Michael Jackson's Vision de Mi... 29,99 US\$ Michael Jackson
	Anthology: The Best of Michae... 29,99 US\$ Michael Jackson
	The Definitive Collection: Micha... 7,99 US\$ Michael Jackson
	Lullaby Renditions of Michael J... 11,99 US\$ Rockabye Baby!
	Unity: The Latin Tribute To Mich... 9,99 US\$ Tony Succar
	VSQ Performs the Hits of Mich... 11,99 US\$ Vitamin String Quartet
	Michael Jackson vs Elvis Presley... 1,29 US\$ Epic Rap Battles of History
	The Instrumental Hits of Michae... 5,99 US\$ Pierre Vangelis

Cancelar

Pero aún hay más. Se puede ir un poco más allá y crear listas personalizadas.



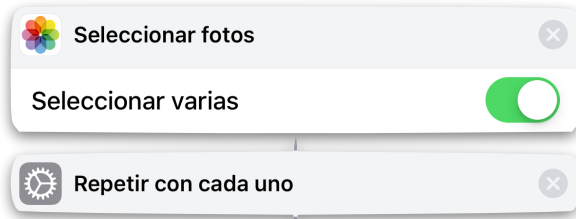
[DESCÁRGALO AQUÍ](#)

Si ya lo has ejecutado (selecciona más de una foto que tenga datos exif), habrás visto que el aspecto de la lista resultante es parecido a esto:

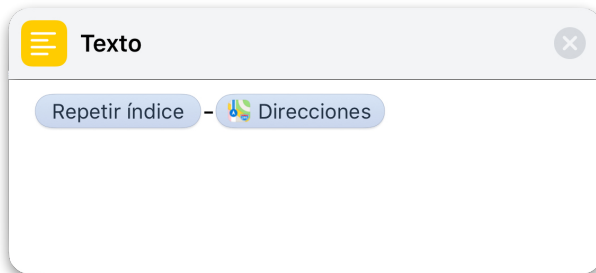


La manera de conseguir esto es usando la técnica siguiente:

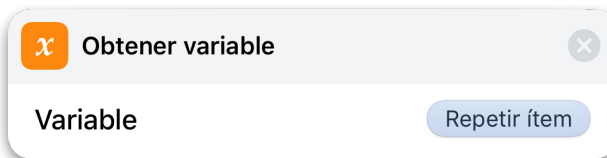
1. Pasar una lista de imágenes a un bloque “*Repetir con cada uno*”



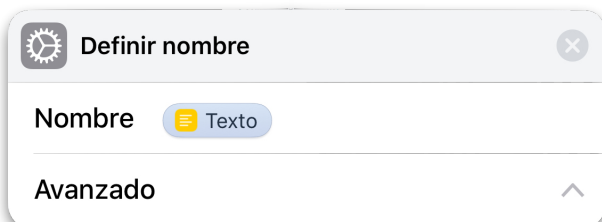
2. Obtener/construir la cadena de texto deseada, en este ejemplo solo se ha añadido un índice, y la dirección.



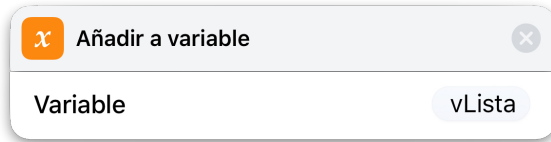
3. "Obtener Variable": Recuperando la imagen del bloque Repetir.



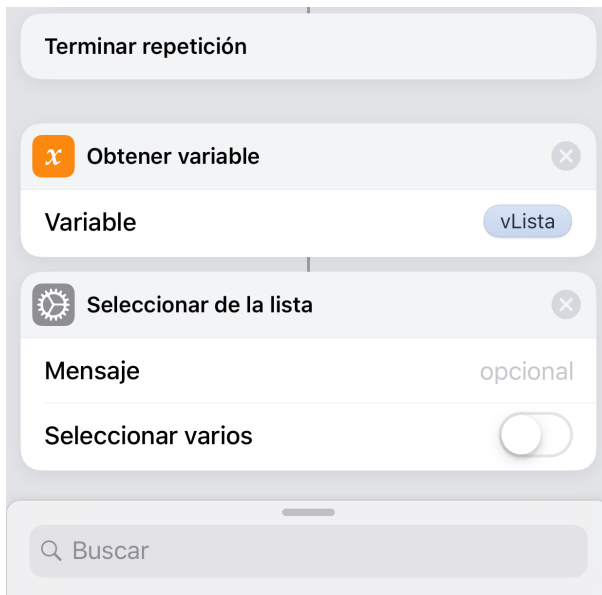
4. "Definir Nombre": Aquí se renombra la foto con el texto construido en el punto 2.



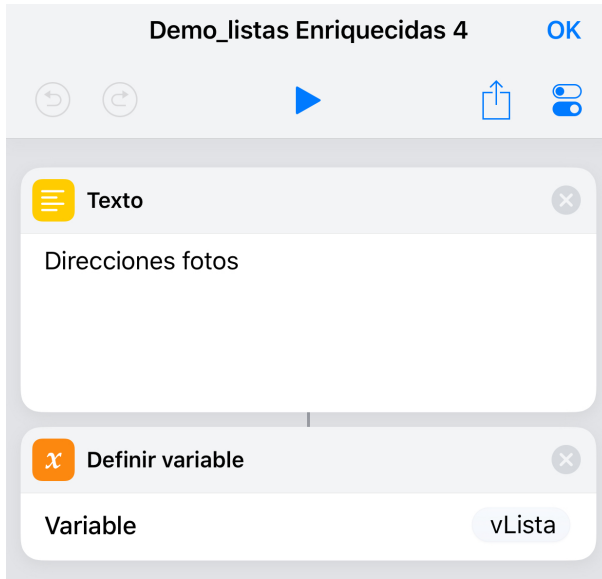
5. “*Añadir a Variable*”: Vamos añadiendo esta foto renombrada con el texto “construido” a la variable “vLista”. Recuerda que en “vLista” se irán añadiendo las distintas fotos ya renombradas.



6. Finalmente, al salir del bloque “*Repetir con cada uno*” ya podrás recuperar a vLista, y pasarla como entrada al bloque “*Seleccionar de la lista*”, obteniendo una lista enriquecida como esta:



Es **muy importante** un detalle que no se ha comentado. Fíjate al principio del shortcut:



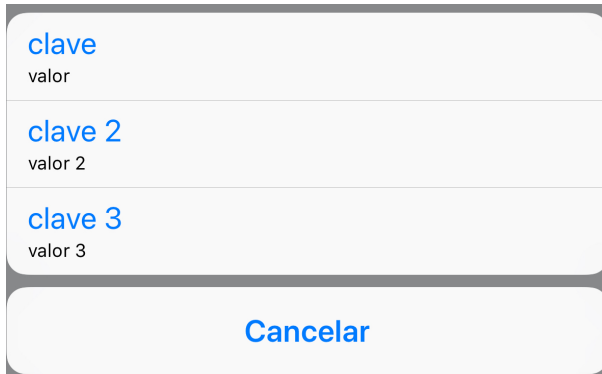
Estos dos bloques son más importantes de lo que parecen. Descarga de [aquí](#) el mismo shortcut anterior, pero sin estos dos bloques. Ejecútalo, y verás que el “*Seleccionar de la Lista*” arroja un resultado un poco distinto. Si estás pensando que Shortcuts ha “pensado” que lo que se está listando son solo fotos, y por eso te las presenta en formato “thumbnail” (vista en miniatura), estás en lo correcto.

Es decir, el bloque “*Texto*” que era la “excusa” como título, lo que hace es añadir a la variable “vLista” una cadena de texto **sin foto**, por lo que Shortcuts **no** presenta los thumbnails ya que tiene dos tipos de objeto, uno con texto, y el resto con las fotos “renombradas”. Por lo tanto, Shortcuts deduce que “quieres” un formato texto con foto en miniatura.

Con todo esto es importante que aprendas dos cosas, a tratar con listas, y sobretodo, intuir un poco más como “piensa” Shortcuts.

Para acabar, una última recomendación. Se pueden conseguir listas enriquecidas del siguiente modo: pasar un diccionario como entrada al bloque “*Seleccionar de la lista*”. Este puede ser formado de forma manual por el propio usuario, o puede venir formado por una llamada a una API. Por ejemplo:





Al pasar un diccionario como entrada al “*Seleccionar de la lista*”, Shortcuts ha usado su “magia” y ha colocado en azul, como título, cada clave del diccionario, y en gris, a modo de subtítulo cada valor.

Con todo esto, y lo que vendrá en el siguiente capítulo, puedes llegar a hacer cosas como está, crear una búsqueda de películas a través de una Web API, y mediante el recorrido de los datos devueltos y un poco de paciencia, crear un "Seleccionar de la Lista" como este:

Escoge una



[Spider-Man \(2002\) id:557](#)

Imagen



[Spider-Man: Homecoming \(2017\) id:315...](#)

Imagen



[Spider-Man: Un Nuevo Universo \(2018\) i...](#)

Imagen



[Spider-Man 3 \(2007\) id:559](#)

Imagen



[The Amazing Spider-Man \(2012\) id:1930](#)

Imagen



[The Amazing Spider-Man 2: El poder de...](#)

Imagen



[Spider-Man 2 \(2004\) id:558](#)

Imagen

[Spiderman \(1990\) id:68658](#)

Texto enriquecido



[Spider-Man: Lejos de Casa \(2019\) id:429...](#)

Imagen



[Italian Spiderman \(2007\) id:112679](#)

Imagen

[Cancelar](#)

OK

<div>



Spider-Man 2

Estreno: 2004-06-25

Director: Sam Raimi

Puntuación: 6,9

Sinopsis: Han pasado dos años desde que el tranquilo Peter Parker dejó a Mary Jane Watson, su gran amor, y decidió seguir asumir sus responsabilidades como Spider-Man. Peter debe afrontar nuevos desafíos mientras lucha contra el don y la maldición de sus poderes equilibrando sus dos identidades: el escurridizo superhéroe Spider-Man y el estudiante universitario. Las relaciones con las personas que más aprecia están ahora en peligro de ser descubiertas con la aparición del poderoso villano de múltiples tentáculos Doctor Octopus, "Doc Ock". Su atracción por M.J. se hace más fuerte mientras lucha contra el impulso de abandonar su vida secreta y declarar su amor. Mientras tanto, M.J. ha seguido con su vida. Se ha embarcado en su carrera de actriz y tiene un nuevo hombre en su vida. La relación de Peter con su mejor amigo Harry Osborn se ha alejado por la creciente venganza de Harry contra Spider-Man, al que considera responsable de la muerte de su padre.

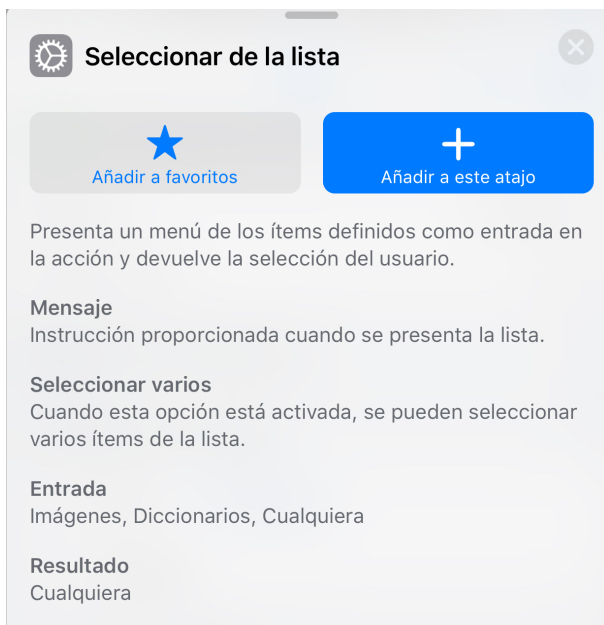


Este último [ejemplo](#) es algo más complejo, y requiere obtener la API KEY de forma gratuita en [TheMovieDB](#)

En esta lista puedes ver cómo se muestra la carátula de la película a la izquierda, y en azul se muestra el título y el id. Esto se consigue usando bloques “*Repetir con cada uno*” y

“Añadir a variable” para “construir” a medida la lista enriquecida.

En definitiva, experimenta y prueba. Si te fijas, en la descripción del bloque “*Seleccionar de la lista*” se puede leer :



Este último “cualquiera” está invitando a probar y a desafiar al usuario para intentar cosas nuevas, así que no defraudes al grupo de desarrollo de Shortcuts.

Un comentario final sobre las listas enriquecidas, úsalas. Las pocas posibilidades de mostrar contenido de una forma atractiva que ofrece la aplicación hay que explotárselas. Una lista de aplicaciones, canciones, o lo que sea que estés “listando” te dará muchas pistas y hará que tu uso de los shortcuts sea con una interfaz, dentro de las limitaciones obvias, mucho más potente.

7.1.2 Listas más enriquecidas con vCard

Gracias a la astucia de los usuarios de Shortcuts ha "aparecido" una técnica más para dotar de una interfaz visual más rica a tus "aplicaciones".

Antes de explicar como se hace y mostrar algún ejemplo, es necesario entender que es una vCard. Una vCard, también conocida como VCF (Virtual Contact File), es un formato standard de fichero que contiene información de un contacto, números de teléfono, email, fotografía e incluso archivo de audio. Las vCards son usualmente adjuntadas a mensajes de correo electrónico, pero pueden ser intercambiadas en muchas otras formas, como en la World Wide Web o a través de códigos QR.

Los archivos que contienen información de tipo vCard llevan extensión .vcf o .vcard.

En Shortcuts los contactos son "parseados" como ficheros .vcf (vCards), y al ser un formato abierto, se puede aprovechar este hecho para crear ficheros .vcf "falsos", engañando a la aplicación y poder crear así una mezcla de gráficos y texto y alimentar, por ejemplo, a un bloque "Seleccionar de la lista".

Esta técnica tiene el único fin de embellecer tus shortcuts, y por contra implica unos bloques más y un poco de trabajo extra. El objetivo final es poder utilizar texto e imágenes en tus bloques "Seleccionar de la lista". Para lograrlo, como se ha dicho, se utilizan las vCards como "contenedores" de la información a mostrar y así se consiguen resultados como los que verás a continuación.

Aplicando este tipo de formato en Shortcuts, se pueden crear de forma manual tarjetas de este tipo y adjuntarlas por ejemplo en un email, o buscar un uso alternativo: Crear listas y menús con un aspecto visual.

Los ingredientes para llevar a cabo esta técnica son:

- Bloque "**Texto**": Es donde se almacena la información, debe seguir un formato determinado.
- Bloque "**Definir Nombre**": Tendrá un nombre del tipo un_nombre.vcf
- Bloque "**Obtener Contactos de la Entrada**": Recoge la información en formato vCard y la pasa como entrada al bloque siguiente.
- Bloque "**Seleccionar de la Lista**": Recibe la vCard y muestra las opciones disponibles.

La clave pues, es crear esta vCard falsa. El formato de las vCard contiene varios campos opcionales, pero para la finalidad que se busca, estos serían suficientes:

```
BEGIN:VCARD
VERSION:3.0
N;CHARSET=utf-8:Texto Que se mostrará;;;
ORG;CHARSET=utf-8:Subtítulo;;;
PHOTO;ENCODING=b:Texto_Foto_Codificada_Base64
END:VCARD
```

Es decir, pegar este fragmento en un bloque "**Texto**" es el primer paso. Para codificar la imagen, puedes, como no, crearte un shortcut que te deje en el portapapeles la imagen reducida. Tienes uno aquí:



[DESCÁRGALO AQUÍ](#)

Nota que la resolución óptima de la foto a añadir debería ser como máximo 123x123 px.

El siguiente paso es un bloque "**Definir Nombre**", donde lo importante es definir la extensión como .vcf.

A continuación "**Obtener Contactos de la Entrada**", que interpreta la vCard y la pasará al bloque siguiente. Finalmente, el texto junto con la imagen codificada es la que aparece como opción en la lista/menú. Lógicamente, si se quieren añadir varias opciones, el formato a seguir es el siguiente:

```
BEGIN:VCARD
VERSION:3.0
N;CHARSET=utf-8:Opción 1;;;
ORG;CHARSET=utf-8:Subtítulo;;;
PHOTO;ENCODING=b:Texto_Foto_Codificada_Base64
END:VCARD
BEGIN:VCARD
VERSION:3.0
N;CHARSET=utf-8:Opción 2;;;
ORG;CHARSET=utf-8:Subtítulo 2;;;
PHOTO;ENCODING=b:Texto_Foto2_Codificada_Base64
END:VCARD
```

Como puedes ver el texto del título en el menú será el colocado en el campo:

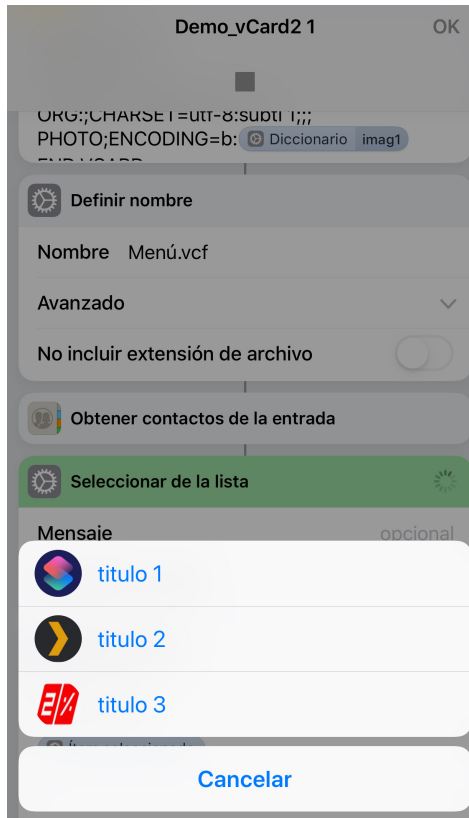
```
N;CHARSET=utf-8:TEXT0;;;
```

Y el subtítulo que aparecerá debajo es el que pongas en la parte:

```
ORG;CHARSET=utf-8:TEXT0;;;
```

La fotografía que aparecerá como icono a la izquierda irá aquí:

PHOTO;ENCODING=b:Texto_Foto_Codificada_Base64



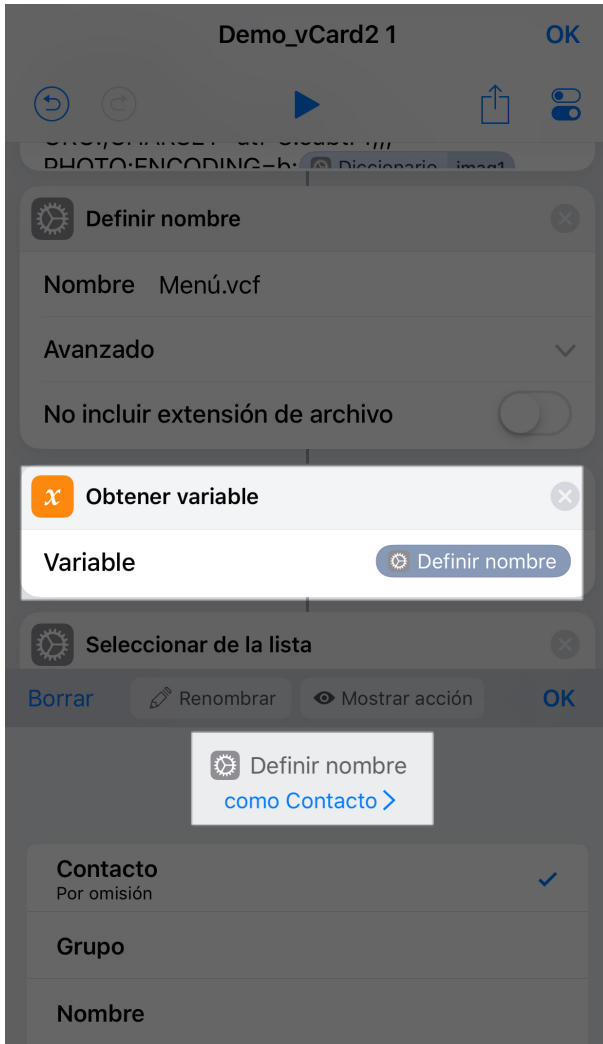
LISTA ENRIQUECIDA

En el enlace siguiente podrás descargar un ejemplo con un simple menú con 3 opciones.



[DESCÁRGALO AQUÍ](#)

Un aspecto que te habrá sorprendido si has instalado el ejemplo es que pide acceso a “contactos”. El motivo es, como se ha dicho anteriormente, que se usa el bloque “Contactos” para “engañar” a la aplicación. Pues bien, para poder sortear este inconveniente y no asustar a los usuarios más desconfiados, hay una solución ¡las variables mágicas! Observa la captura y mira como se utiliza un “Obtener Variable” y se realiza un “cambio de tipo de variable” definiéndola como Contacto.



VARIABLE MÁGICA Y CAMBIO DE TIPO



[DESCÁRGALO AQUÍ](#)

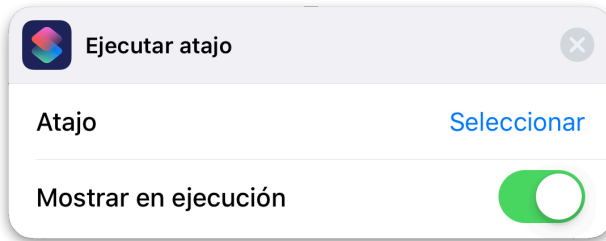
Un último apunte, como habrás podido ver en ambos ejemplos, lo más fácil es crear un diccionario inicial y añadir allí las imágenes codificadas. De este modo es mucho mas sencilla la creación del bloque "Texto" con el formato .vcf

7.2. Flujos de ejecución

A la hora de ejecutar un shortcuts hay varias opciones como pudiste ver en el capítulo 2.3 del libro.

Sin embargo, conforme se avanza en la creación de shortcuts complejos, puede que la ejecución lineal, tal y como está planteada en la aplicación no sea suficiente, y sea necesario llamar a otro shortcut desde el que estamos ejecutando para que realice cierta tarea sencilla. Normalmente porque se divide en shortcuts mas pequeños que pueden ser llamados desde otros o, lo que es lo mismo, se separa una acción en un shortcut para porque esa acción en concreto la usamos en varios shortcuts.

Un ejemplo es aquel en el que tienes un shortcut que recopila cierta información sobre lo que has corrido en tu ultima carrera, km, velocidad etc, y una vez recogida toda esa información te planteas que quieres compartirla en twitter, y ya tienes un atajo para compartir información en twitter. En ese caso no es necesario que vuelvas a repetir todas las acciones, es mejor llamar a ese shortcut mediante la acción ejecutar atajo.



ACCIÓN EJECUTAR ATAJO

En esta acción se pueden configurar dos elementos, por un lado, el shortcut que se va a llamar y por otro si se quiere que la ejecución sea transparente (en segundo plano) o que se abra el atajo para que se vayan viendo las acciones durante la ejecución.

Con respecto al término transparencia, estamos acostumbrados a que en el lenguaje del día a día, la transparencia se asocie a mostrar todo para que se sepa que no hay nada escondido: transparencia en las cuentas, transparencia en las acciones (sobre todo aplicados a la política). Pero, cuando algo es transparente, indica que se puede ver a través de él, dicho de otra manera, es invisible.

En informática la transparencia se usa para esto precisamente, un proceso transparente es aquel que ocurre sin que el usuario se percate o lo moleste.

De cara a la velocidad de ejecución y a la carga de trabajo en el dispositivo siempre es mejor ejecutar de manera transparente, ya que eso no elimina los avisos o peticiones de información necesarios para la ejecución.

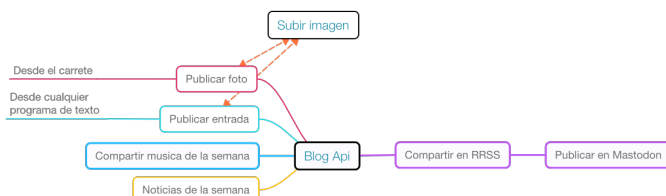
El shortcut que se haya decidido abrir recibirá la información que esté en el hilo de ejecución del shortcut que estemos ejecutando y continuará tratando esos datos. Cuando termine volverá al atajo original que continuará con los datos devueltos por el segundo atajo como si fueran propios, por lo

que no hay que preocuparse del intercambio de información entre ellos.

Dicho de otro modo, el hilo de información no se interrumpe al cambiar de shortcut, y esta es tratada como si todo se estuviera procesando dentro de un solo shortcut mas largo.

7.3. Reutilización de código en atajos

Al hilo del capítulo anterior, a continuación puedes ver un ejemplo de un conjunto de shortcuts que funcionan conjuntamente, es un ejemplo de real de unos shortcuts para publicar en un blog que funciona usando la API de [BludIt](#).



FLUJO DE SHORTCUTS PARA PUBLICAR EN BLOG

En el gráfico se ve cómo hay cuatro tipos de entradas que se manejan mediante shortcuts diferentes. Algunos de ellos necesitan subir imágenes al blog, y para ello se ha creado un shortcut que sube las imágenes por webdav y devuelve la url de la imagen para insertarla. Este sub-shortcut, por llamarlo de alguna manera, es llamado tantas veces como imágenes haya en la entrada.

Una vez que se ha formateado toda la entrada según lo requiere el shortcut que gestiona la API se le pasa la información y éste la sube al blog, y al terminar devuelve la URL de la entrada recién publicada. Esta url se le envía a otro

shortcut que gestiona las redes sociales y avisa a los seguidores de que hay una nueva entrada en el blog.

Dentro de este shortcut se publica automáticamente en twitter y Facebook, pero [mastodon](#), por ahora no está soportado, de ahí que se llame a un shortcut externo que se ha creado para publicar en mastodon.

A priori todo esto puede resultar complejo, no fue así como se planificó en un principio, pero a medida que vas cubriendo tus necesidades te vas dando cuenta de que hay partes del código de un shortcut que necesitarías usarlo más veces y ahí es cuando decides empezar a separar el código para reutilizarlo de forma más eficiente. Esto es el equivalente a las funciones en programación.

Si ya con shortcut buscamos automatizar tareas repetitivas, con la reutilización de código se le da una vuelta más y se automatizan tareas repetitivas dentro de las tareas repetitivas.

No tiene sentido que si vamos a publicar en una red social en muchos shortcuts programemos una y otra vez la publicación, es mucho más óptimo hacerlo una vez bien, haciendo que ese sub-shortcut admita lo que se va a publicar de una determinada manera, la que creas mejor, y a partir de ahí el resto de shortcuts usen ese sub-shortcut para publicar en la red social.

Se puede decir, que la reutilización de código es uno de los puntos más importantes para un programador, y que incluso se juzga lo bueno o malo que es un programa en función de como de bien se haya planificado esta reutilización, así que intenta tomártelo como un reto y verás como la importancia no es arbitraria, viene dada porque realmente ayuda a ser más eficientes y a cometer muchos menos errores.

Imagina un shortcut largo, en la que se haga lo mismo varias veces en distintos puntos del shortcut, descubrir un error para arreglarlo es un autentico problema, más aún si ha pasado un

tiempo desde que has creado ese shortcut o si ni siquiera lo has creado tu porque es un shortcut que has bajado de internet.

Es verdad que esto añade más shortcuts a tu galería y por tanto puede llegar a ensuciarla. Para ello, mientras Apple no ofrezca una forma de crear carpetas para agrupar los shortcuts de alguna manera, lo más eficiente es usar los colores para diferenciar bien los shortcuts que hacen algo, o aquellos que son funciones concretas que se llaman desde otros shortcuts.

También puedes usar un prefijo en el nombre, por ejemplo [sub] subir imagen para un sub-shortcut que sube imágenes. Al final será la necesidad la que te haga ser creativo a la hora de organizarte.

Véase un ejemplo: Imagina que tienes un conjunto de direcciones que usas habitualmente, y que además las usas en varios shortcuts. La mejor manera de planterlo sería crear un fichero de config, y recuperarlo usando un shortcut que haga específicamente esa tarea.

Pasos:

1. Crear fichero de configuración
2. Crear Subshortcut. ([Sub Destinos1](#))
3. Crear Shortcut Principal([Demo subshortcut](#))

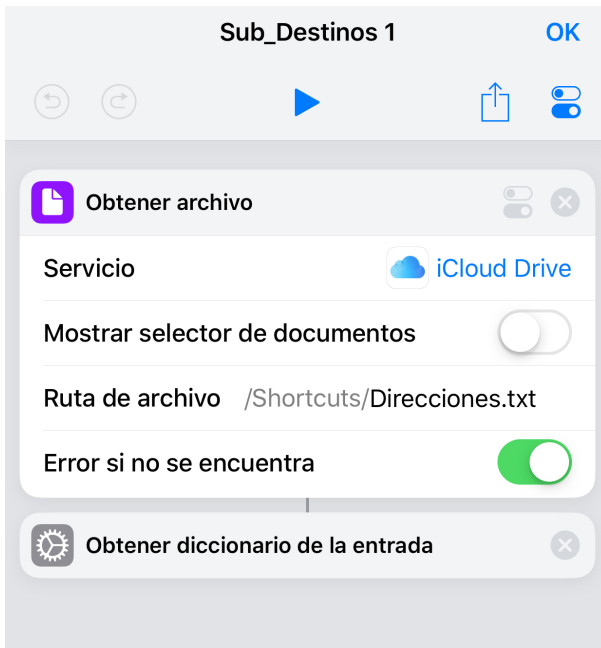
En primer lugar, abre uno de los editores de texto recomendados anteriormente, y crea un fichero nuevo pegando lo siguiente:

```
{ "Casa":"mi+casa", "trabajo":"calle+de+tu+casa,madrid",  
" M a m a " : " s u + d i r e c c i o n , s e g o v i a " ,  
"amigo":"Carril,san+Ginés,murcia" }
```

Como deberías saber ya, es un fichero JSON, verás más adelante porque es preferible utilizar JSNO y no texto plano separado por algún carácter como “-”.

Guarda este fichero con el nombre “Direcciones.txt” en la carpeta de iCloud>Shortcuts.(Respeta la “D” mayúscula)

Ya se ha creado un fichero de configuración, ahora crea el subshortcut, que no es más que un shortcut en toda regla, pero que “devolverá” información:

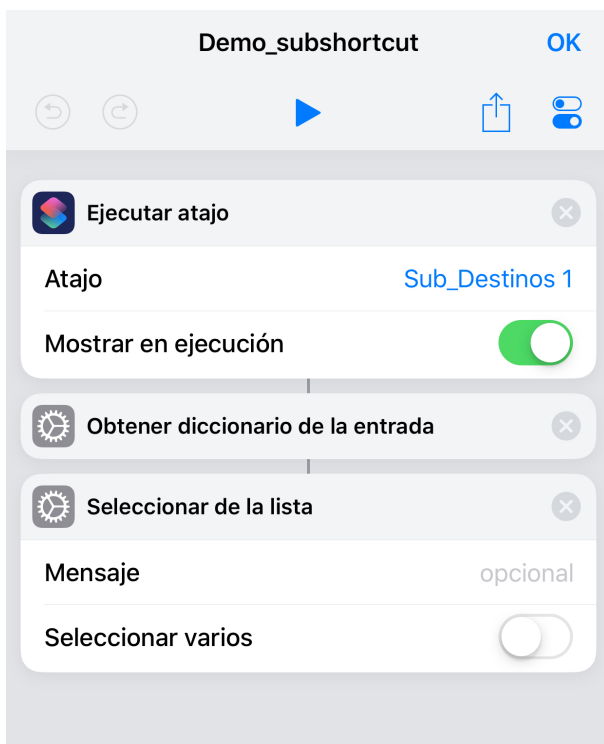


SUBSHORTCUT



[DESCÁRGALO AQUÍ](#)

Es un simple shortcut que recupera el fichero de configuración. Paso final, crear el shortcut principal:



SHORTCUT PRINCIPAL

Si ejecutas este último, verás un resultado del tipo:



RESULTADO

7.4. Diccionarios

Los diccionarios están estrechamente relacionados con las listas, ya que son una evolución de las mismas; y se

relacionan también con el formato JSON en cuanto a que éste es el formato en que se guardarán las listas en Shortcuts.

Los diccionarios son una lista de pares de valores en los que el primero es la clave y el segundo el valor:

```
"clave": "valor"
```

La **clave** será un valor de texto (string) que se usará para recuperar lo que se haya almacenado en el campo "valor".

Conviene aclarar que los diccionarios son una lista **no ordenada**, es decir, el orden en el que se añadan los datos no es el orden en el que se almacenan. Se almacenan de forma aleatoria, tampoco es alfabético. Aunque hay una excepción a este tema que se aclarará mas adelante.

A continuación observa este ejemplo de diccionario:

```
{  
  "color": "negro",  
  "nRuedas": 4,  
  "nPuertas": 5,  
  "encendido": false,  
  "faros": ["posición", "cortas", "largas",  
  "antiniebla"]  
}
```

¡Sí! es el mismo que se estudió en el capítulo de JSON. Las claves, en este caso, son: color, nRuedas, nPuertas, encendido y faros.

Los valores son: "negro", 4, 5, false y ["posición", "cortas", "largas", "antiniebla"].

Los diccionarios, en su representación en texto siempre van enmarcados por unas llaves, y sus pares de valores separados por , excepto el último par de clave/valor, aunque si este llevara una coma se interpretaría correctamente.

Puedes ver que los valores pueden almacenar cualquier tipo de dato: texto, número, booleano o lista, otro diccionario, imagen...

Las acciones que se usan para trabajar con diccionarios en Shortcuts son:

- **Diccionario:** crea un diccionario introduciendo los valores manualmente
- **Ajustar valor del diccionario:** asigna un valor a una clave dada.
- **Obtener valor del diccionario:** obtiene un valor, todos los valores o todas las claves de un diccionario.
 - Si obtiene **un valor**, este tendrá el formato original, texto si era texto, número si era número, etc.
 - Si obtiene **todos los valores**, se obtendrá una lista en la que estarán todos los valores que haya en el diccionario.
 - Si obtiene **todas las claves**, el resultado será una lista con todos los valores del diccionario.
- **Obtener diccionario de la entrada:** coge los datos de entrada y los guarda en un diccionario, siempre que sea posible porque los datos estén correctamente formateados, si no, dará error.

7.4.1. Crear diccionarios

Shortcuts admite varias formas de crear diccionarios. Cada una con sus ventajas e inconvenientes.

7.4.1.1. Crear un diccionario de forma manual

Usando la acción “Diccionario” puedes crear un diccionario de forma manual introduciendo los datos (claves y valores) tú mismo.

Icono	Clave	Valor	Acción
−	miClave1	mi valor 1	☰
−	miClave2	2	☰
−	Clave	Texto	☰
−	Clave	0	☰
−	Clave	2 ítems	☰
−	Clave	1 ítem	☰
−	Clave	Falso	☰
+	Añadir nuevo ítem		

ACCIÓN DE CREACIÓN DE DICCIONARIO DE FORMA MANUAL CON DATOS DE EJEMPLO.

Tal y como puede verse en el ejemplo hay distintos tipos de datos que se pueden introducir: * Texto * Numero * Matriz * Diccionario * Booleano

¿Qué tipo de valor quieres añadir?

Texto

Número

Matriz

Diccionario

Booleano

TIPO DE DATOS QUE SE PUEDEN AÑADIR EN LOS DICCIONARIOS.

Seleccionar el tipo de dato que se va a introducir no es más que una ayuda para la introducción de datos, ya que los diccionarios admiten solo texto y son los programas los que tienen que reconocer el tipo de dato que se recupera.

Esto no quiere decir que no deban usarse estas ayudas, al contrario, están ahí para hacer la introducción de datos más sencilla.

Este tipo de diccionarios, en los que los datos se introducen de forma manual, se usan habitualmente para colocar al principio la configuración del shortcut. Datos como usuario, contraseña, ruta de algún archivo, url o cualquier otra cosa que pueda necesitar tu shortcut.

7.4.1.2. Crear un diccionario a partir de texto

Es posible preparar un texto para que Shortcuts lo convierta en un diccionario, para ello, éste deberá tener un formato adecuado.

Los formatos que se pueden usar son dos: * JSON *
clave=valor&

En ambos casos deberá usarse la acción “Obtener diccionario de la entrada” para conseguir el diccionario.

El primero está estudiado, por lo que no necesita mas aclaraciones, si tienes un texto con formato JSON y usas la acción “Obtener diccionario de la entrada” automáticamente se almacenarán los datos en el diccionario de forma correcta.

El segundo es el más interesante de todos, ya que es el único que permite obtener un **diccionario ordenado**. Es decir, los datos se almacenarán en el orden en el que se le hayan proporcionado.

El esquema es el siguiente: clave1=valor1 separado del siguiente valor por el símbolo ampersand &.

El ejemplo anterior quedaría formateado de esta manera:

```
color=negro&nRuedas=4&nPuertas=5&encendido=fal  
se&faros=["posición", "cortas", "largas",  
"antiniebla"]
```

Como puedes ver, la capacidad para introducir datos mediante este formato es la misma, con la ventaja de obtener un diccionario ordenado, lo cual puede ser útil en algunos casos.

7.4.1.3. Crear un diccionario a partir de una url

Puede que tengas una API, que se verán en el capítulo siguiente, o una web que arroje los resultados en formato JSON y necesites usarlos en tu shortcut.

En ese caso en una caja de acción del tipo “URL” introduce la dirección completa de la web que arroja el resultado JSON.

A continuación usa una acción “Obtener contenido de la URL” para hacer que Shortcuts recupere la información de la web, que en este caso será un texto con el JSON.

A partir de aquí, los pasos son los mismos que en el caso de un diccionario a partir de un texto, usa una acción “Obtener diccionario de la entrada” y lo tendrás todo listo para empezar a trabajar con él.

7.4.1.4. Crear un diccionario a partir de un archivo

Si tienes un archivo txt con información estructurada en formato JSON o un archivo .json puedes usar la acción “Obtener archivo” para que tu shortcut lo seleccione desde iCloud o Dropbox y lo cargue. De nuevo, con una acción del tipo “Obtener diccionario de la entrada” tendrás todo listo para poder trabajar con el diccionario.

7.4.2. Leer valores de un diccionario

7.4.2.1. Leer valores de un diccionario

Sea cual sea el origen de tus datos, texto, un archivo, una url o que lo hayas creado manualmente, al final lo que tienes es una acción `Diccionario` que contiene los datos de forma estructurada **clave/valor** pero, recuérdalo, desordenada.

Mediante la acción “Obtener valor del diccionario” puedes acceder a los datos del diccionario de forma sencilla.

De nuevo, supón que tienes el diccionario del ejemplo:

```
{
  "color": "negro",
  "nRuedas": 4,
  "nPuertas": 5,
  "encendido": false,
  "faros": ["posición", "cortas", "largas",
"antiniebla"]
}
```

Y necesitas obtener el valor almacenado en la clave “color”. Usarás la acción “Obtener valor del diccionario” con la siguiente configuración: * Obtener: valor * Clave: color

Debido a que las claves son cadenas de texto, es muy importante que te asegures de que la has escrito de forma correcta: mayúsculas, minúsculas, acentos, espacios,... cualquier diferencia con el nombre exacto del valor hará que la caja de acción no encuentre el dato que buscas.

Al ejecutar el shortcut con esta configuración obtendrás negro como resultado.

Se da con cierta frecuencia, aunque ahora mismo pueda parecerle enrevesado o remoto, que haya un diccionario

dentro del valor de una clave. Se usaba este ejemplo en el capítulo sobre JSON:

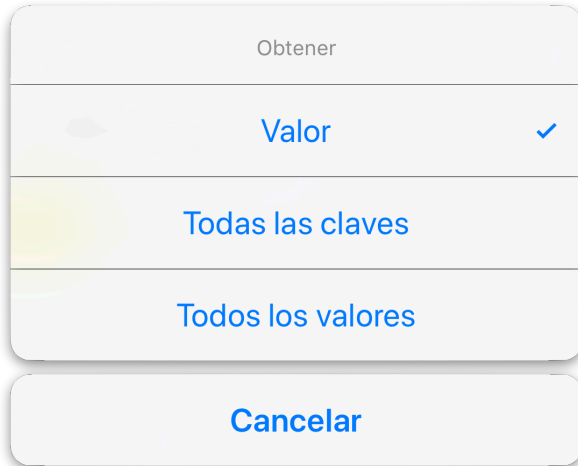
```
{
  "color": "negro",
  "nRuedas": 4,
  "nPuertas": 5,
  "encendido": false,
  "faros": ["posición", "cortas", "largas",
"antiniebla"],
  "extras": {
    "control de cruce": true,
    "airbags": ["conductor",
"acompañante", "laterales"],
    "infotainment": {
      "pantalla": 7,
      "CarPlay": true,
      "radio": ["am", "fm", "DAB"]
    }
  }
}
```

La clave “extras” contienen un diccionario donde se han almacenado los extras del vehículo virtual.

En este caso, el resultado de “Obtener valor del diccionario” nos arrojará otro diccionario. Dicho de otro modo, habrás entrado en el segundo nivel de este diccionario. Si solo necesitas obtener un valor de este segundo nivel puedes volver a usar una caja “Obtener valor del diccionario” con alguna clave existente en este segundo diccionario y el programa te lo dará en la salida.

Si necesitas trabajar con más profundidad en el mismo es mejor que uses una acción “Obtener diccionario de la entrada” para estructurarlo bien y guardarlo en una variable que te permita recuperar sus datos cuántas veces quieras.

Esta acción tiene, además otras opciones cuando pulsas en Obtener Valor en azul. Aparecerá un menú desplegable con las opciones: * Valor * Todas las claves * Todos los valores



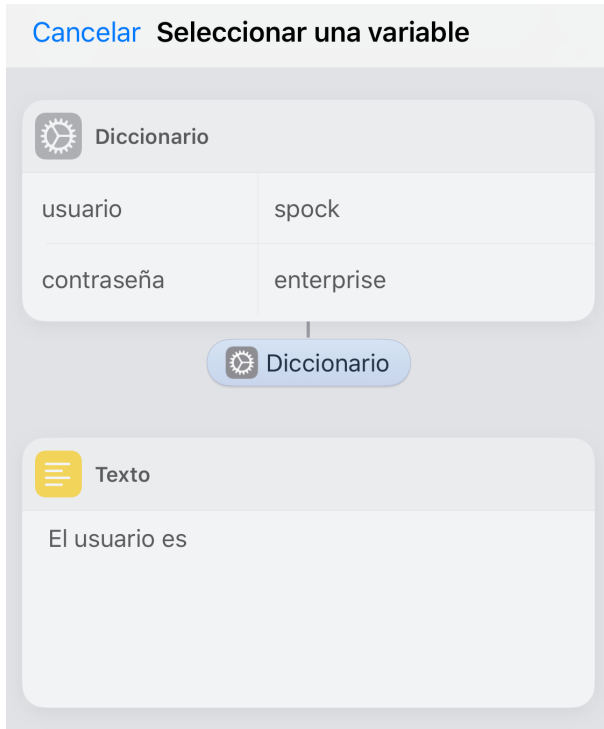
OPCIONES DISPONIBLES AL PULSAR SOBRE “VALOR”

La opción de “Todas las claves” te devolverá una lista con todos los textos que se han usado como clave.

La opción de “Todos los valores” te devolverá una lista con todos los valores que se han almacenado en todas las claves.

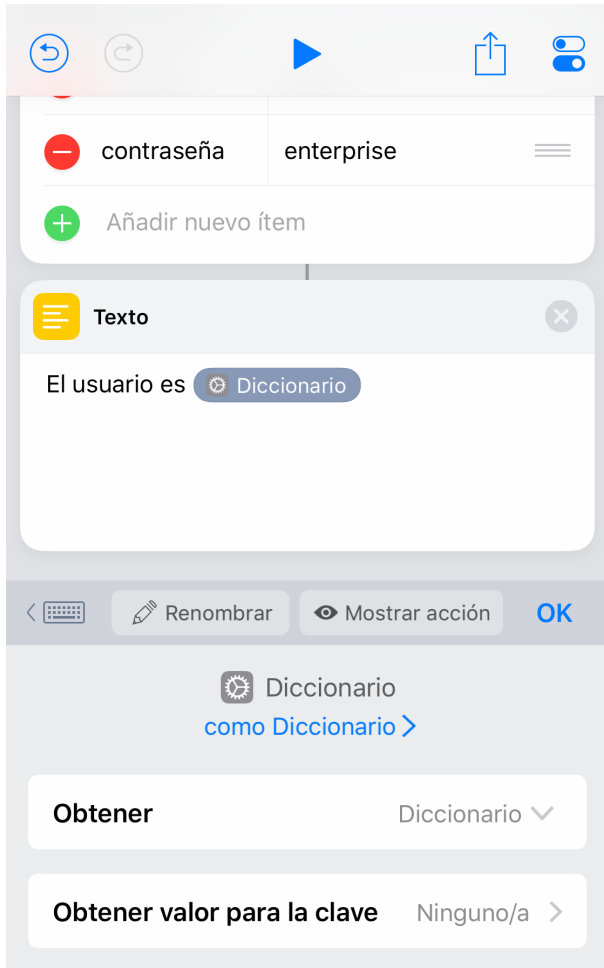
Recuperar un valor sin acciones

Hay una manera que optimiza mucho la cantidad de acciones que se deben usar a la hora de trabajar con Diccionarios que consiste en acceder al diccionario mediante una variable mágica o la variable que hayas creado para contenerlo.



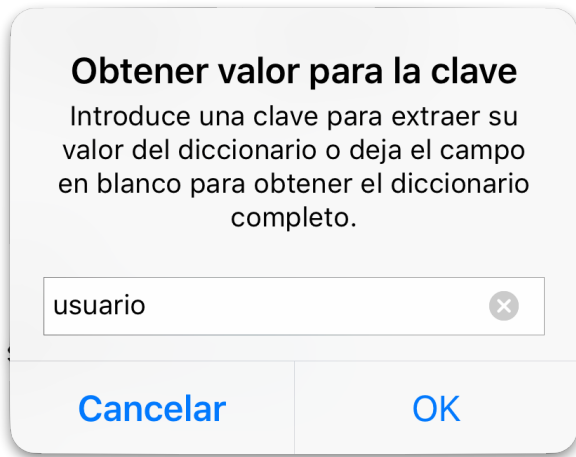
SELECCIÓN DE LA VARIABLE MÁGICA

Una vez introducida pulsa sobre la variable y aparecerá un menú de opciones.



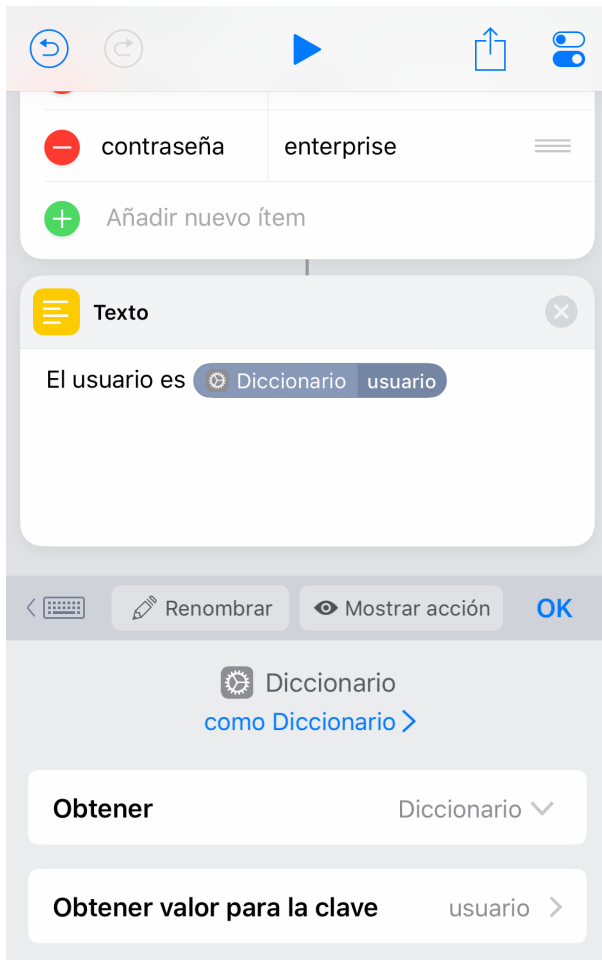
OPCIONES AL SELECCIONAR UNA VARIABLE QUE ES UN DICCIONARIO

Observa que abajo del todo aparece una opción para introducir una clave. Si haces clic sobre ella se abre un cuadro de dialogo en el que puedes escribir el texto de dicha clave.



CUADRO DE DIÁLOGO PARA INTRODUCIR EL NOMBRE DE LA CLAVE QUE VAS A RECUPERAR

Al aceptar verás como ahora, junto a la variable aparece el nombre de la clave que has escrito, por lo que, a partir de este momento lo que has recuperado no es el diccionario completo. Has recuperado solo el valor de la clave que aparece junto al diccionario.



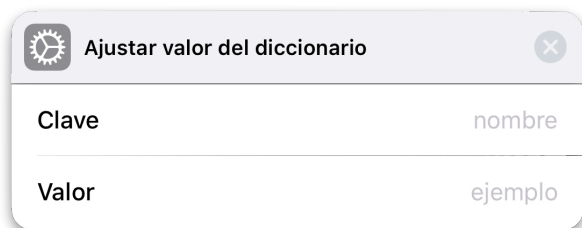
LA VARIABLE MÁGICA AHORA MUESTRA LA CLAVE QUE SE OBTENDRÁ

7.4.2.2. Escribir valores en un diccionario

A menudo, si haces un shortcut que guarde información de forma permanente en Dropbox o iCloud, necesitarás guardar

ciertos valores en una clave dentro de un diccionario que al final se guardará en el disco duro de tu dispositivo.

Para ello, de nuevo, partirás del diccionario que se ha obtenido por cualquiera de los métodos mencionados anteriormente y se usará la acción “Ajustar valor del diccionario”.



Ajustar valor del diccionario	
Clave	nombre
Valor	ejemplo

ACCIÓN “AJUSTAR VALOR DEL DICCIONARIO”

En el campo “Clave” escribe la clave que ya exista para ese valor, en cuyo caso, el valor se sobrescribirá; o una clave nueva, en este caso se añadirá al diccionario la nueva clave y el valor que le acompañe.

En el campo “Valor” introduce el valor que deseas almacenar. Como bien sabes a estas alturas puede ser un texto o número escrito de forma manual, una variable o una variable mágica.

7.4.3. Ejemplo sencillo

Como ejemplo resumen de este capítulo se va a crear un diccionario en el que se guardarán los datos de configuración del shortcut para que sea más fácil de configurar por parte de quién quiera usarlo.

Para ello, en primer lugar añade una acción “Diccionario”. Y añade manualmente los siguientes pares de clave/valor: * usuario: spock * clave: enterprise * url: <http://www.startrek.com>






ACCIÓN DICCIONARIO CONFIGURADA

Ahora, recuperarás en una caja de texto todos los campos usando las variables mágicas.

El resultado será este:

Diccionario		✕
− usuario	spock	☰
− contraseña	enterprise	☰
− url	http://www.startre...	☰
+ Añadir nuevo ítem		

Texto ✕

El usuario es  Diccionario usuario cuya
contraseña es  Diccionario contraseña en la
web  Diccionario url

```
El usuario es spock cuya
contraseña es enterprise en la
web http://www.startrek.com
```

⏪ ⏩ ⏴ ⏵ 

RESULTADO TRAS LA EJECUCIÓN DEL SHORTCUT

Como ves, has podido recuperar cada valor guardado sin mayor problema. Deberías usar este diccionario de configuración para aquellas partes del shortcut que contienen datos que cambiarán y que se usan en muchas otras acciones. De esta manera haciendo el cambio en el diccionario de configuración este se propagará a todo tu shortcut, por lo que te ahorrarás tener que revisarlo de arriba abajo tras un cambio.



[DESCÁRGALO AQUÍ](#)

7.4.4 Novedad Shortcuts 2.2 - Leer clave/valor de diccionario anidado

Una de las tareas sin duda más tediosas cuando se trabaja con diccionarios se da cuando por necesidad se desea recuperar algún valor o clave en un nivel muy inferior. Por ejemplo, imagina un vector que a su vez contenga un diccionario en su interior y quieras recuperar un valor de esta diccionario anidado.

Hasta ahora había que repetir el conjunto de acciones **“Obtener Valor del Diccionario”**, técnica que además de alargar innecesariamente el shortcut, era muy poco eficiente.

Imagina esta estructura:

```
{
  "familia":
    [ "Diccionario1"
      { "hijo1": "frank",
        "hijo2": "rafa",
        "hijo3": "jose"
      }
    ]
}
```

```
        }  
    "Diccionario2"  
        {"suegro1": "Darth Vader"  
        }  
    ],  
}
```

Es decir, un diccionario con una clave "familia" que contiene un vector con 2 diccionarios más, el primero con 3 clave/valor, y el segundo con una única clave/valor.

La estructura comentada sería de este modo en Shortcuts:



familia		OK
-	3 ítems	☰
-	1 ítem	☰
+	Añadir nuevo ítem	

Diccionario		OK
-	hijo1 frank	☰
-	hijo2 rafa	☰
-	hijo3 jose	☰
+	Añadir nuevo ítem	

Importante destacar el aspecto diferencial entre VECTOR y DICCIONARIO, ya que la manera de acceder a los datos diferencia entre uno y otro.

Dicho esto, ahora con un sólo bloque **“Obtener Valor del Diccionario”** podrás recuperar una clave o valor. La nomenclatura es relativamente sencilla (separar cada nivel/vector con “.”) y tener en cuenta que los vectores se numeran empezando por 1.



Por ejemplo, para acceder al hijo2 en la estructura anteriormente comentada sería:

“Obener Valor del Diccionario”

Obtener: Valor

Clave: familia.1.hijo2

Puedes descargar el ejemplo aquí:



[Descárgalo aquí](#)

7.4.5 Diccionarios - Actualizar valores de forma persistente.

Entre las acciones típicas que se realizan con diccionarios se pueden diferenciar dos modos:

Las operaciones de lectura: simplemente implican que a partir de un diccionario obtenido a través de la descarga de un JSON, o un diccionario definido en el propio shortcut, etc; es relativamente sencillo recuperar algún valor, o incluso realizar un bucle para recorrer todas sus claves/valores.

- Las operaciones de lectura/escritura: además de leer un fichero JSON, en ocasiones se requiere poder actualizar valores de estos ficheros. Para hacerlo, además de leer en primer lugar los datos del JSON, después de actualizar uno o más valores estos deben volver a almacenarse para que los datos persistan.

El primer caso ha sido utilizado en numerosos ejemplos a lo largo de este libro. El segundo, en cambio, tiene alguna peculiaridad que merece ser explicada.

7.4.5.1 Actualización de un valor de un diccionario: un nivel.

El primer ejemplo es el caso más sencillo. El proceso es el siguiente:

- Lectura de un JSON que está en la nube y que será descargado en local durante la ejecución del shortcut.
- Posteriormente se hará una modificación a un valor.
- El proceso finaliza con la actualización del fichero en la nube y guardar así el cambio realizado.

Se tomará como base el fichero JSON llamado Demo_Diccionario_.JSON, que estará almacenado en la nube de iCloud\Shortcuts. (el ejemplo que encontrarás en la parte final de este capítulo ya creará el fichero por ti la primera vez que ejecutes el shortcut)

Su estructura :

```
{  
  "color": "negro",  
  "nRuedas": 4,  
  "nPuertas": 5,  
  "encendido": false,  
  "faros": ["posición", "cortas", "largas",  
  "antiniebla"],
```

```

"extras":
{
    "control de crucero": true,
    "airbags": ["conductor", "acompañante",
"laterales"],
    "infotainment":
    {
        "pantalla":
            {
                "tamaño": [7,9,12],
                "resolucion": "300dpi",
                "borde": "3mm"
            },
        "CarPlay": true,
        "radio": ["am", "fm", "DAB"]
    }
}
}

```

Como queda patente, hay un diccionario principal, con varias entradas Clave/Valor. A su vez, hay un segundo diccionario dentro de la clave “extras”, un tercero dentro de la clave “infotainment”, y finalmente un cuarto diccionario dentro de “pantalla”.

Por otro lado, hay varios arrays, como lo son la clave “airbags”, “faros” o “radio”.

En este primer ejemplo se va a suponer que se desea cambiar el valor de la clave “**nPuertas**”. (El enlace al shortcut lo encontrarás al final de esta sección.)

En primer lugar el shortcut intenta la descarga del fichero JSON, y si no existe, lo crea.

A continuación se muestra un aviso indicando el valor actual de “**nRuedas**”, y seguidamente pregunta al usuario cuál va a ser el nuevo valor.

A partir de aquí se llega a la parte interesante:

1. Se recupera la variable “**Diccio**”, que contiene el diccionario
2. Se utiliza el bloque “**Ajustar Valor del Diccionario**”, donde se indica la clave (“nRuedas”), y el nuevo valor (el que el usuario ha introducido anteriormente).
3. Importante: “**Definir Variable**”, es decir, se actualiza la variable “**Diccio**” con las clave/valor con el diccionario ya modificado.
4. “**Definir Nombre**”: Lo que se va a hacer es sobrescribir el fichero creado al inicio. Para hacerlo debe tener lógicamente el mismo nombre.

Con estos pasos conseguirás poder actualizar valores en diccionarios, siempre que las claves a actualizar estén en el nivel superior, es decir, no estés cambiando valores que están en diccionarios anidados.

Puedes descargar el ejemplo aquí:



[Descárgalo aquí](#)

La siguiente pregunta que deberías formularte es: ¿Y cómo se actualizan valores que estén anidados? En el siguiente apartado la respuesta.

7.4.5.2 Actualización de un valor de un diccionario que está anidado. (un nivel)

Cuando se dispone de un JSON con diccionarios anidados y se quiere cambiar el valor de alguna clave que está dos o más niveles por debajo del diccionario principal entonces la estrategia se complica.

Una vez actualizado el valor de la clave deseada, si este valor está dentro de un diccionario anidado, se deben actualizar los diccionarios de niveles superiores para que el diccionario padre esté correcto y este pueda guardarse con las modificaciones realizadas.

En el momento de escribir estas líneas **Shortcuts 2.2** no permite utilizar el bloque "**Ajustar Valor del Diccionario**" del mismo modo que "**Obtener Valor del Diccionario**", de manera que no es posible actualizar valores utilizando la nomenclatura "valor.valor2.valor3" para llegar a valores que están anidados en sub diccionarios.

Siguiendo con el mismo ejemplo anterior, para realizar una modificación del valor "**Control de crucero**" (que está dentro del diccionario "**extras**") se deben realizar los siguientes pasos:

1. Se recupera el fichero JSON (se habrá creado con el ejemplo anterior)

2. Se muestra el valor actual de la clave "**control de crucero**" utilizando la nueva técnica para acceder a valores en Shortcuts 2.2. Utilizando el bloque "**Obtener Valor del Diccionario**" con la expresión "extras.control de crucero" se obtiene el valor actual.

3. Se solicita al usuario el nuevo valor.

A partir de este punto empieza lo realmente importante:

4. Se "bajará" de la forma convencional hasta el valor a modificar usando los bloques "Obtener Valor del Diccionario", pero con una salvedad, a cada nivel que se vaya accediendo se definirá un "**Definir Variable**" para poder utilizarlo después y hacer el trabajo más sencillo.

"Obtener Variable Diccio"

"Obtener Valor del Diccionario" -> extras

"Definir Variable" -> Nivel1

"Ajustar Valor del Diccionario" -> "Control de crucero" se actualiza al valor preguntado anteriormente.

"Definir Variable" -> Nivel1 (Nivel1 contiene la modificación realizada)

Una vez realizada la modificación del valor, ahora hay que aplicar los cambios hacia "arriba":

"Obtener Variable" -> Diccio

"Ajustar Valor del Diccionario"-> "extras" se actualiza al valor de la variable Nivel1

"Definir Variable" -> Diccio

"Guardar Archivo" -> se sobre escribe el fichero que hubiera anteriormente.

El ejemplo se puede obtener en:



[Descárgalo aquí](#)

7.4.5.3 Actualización de un valor de un diccionario que está anidado.(dos o más niveles)

Para finalizar este apartado, se va a mostrar un ejemplo un poco más complejo, actualizando esta vez el valor de la clave **extras-> infotainment -> Pantalla -> resolución**, cuyo valor inicial es de 300dpi, y se va a modificar por el valor que el usuario desee.

Este ejemplo es exactamente igual que el anterior, pero con dos niveles más de profundidad, lo que implica más acciones.

Se omiten los primeros bloques, donde resumiendo se recupera el diccionario de la nube de iCloud, se muestra el valor actual de la clave "resolucion", y se pide al usuario el nuevo valor.

A partir de este momento se "desciende" nivel a nivel hasta llegar al valor a modificar:

"Obtener Variable" -> Diccio

"Obtener Valor del Diccionario" -> extras

"Definir Variable" -> Nivel1

"Obtener Valor del Diccionario" -> infotainment

"Definir Variable" -> Nivel2

"Obtener Valor del Diccionario" -> pantalla

"Definir Variable" -> Nivel3

"Ajustar Valor del Diccionario"-> "resolucion"
se actualiza al valor introducido por el usuario

"Definir Variable" -> Nivel3

Ahora sólo queda actualizar los distintos diccionarios, de
abajo hacia arriba:

"Obtener Variable" -> Nivel2

"Ajustar Valor del Diccionario"-> "pantalla" se
actualiza a la variable Nivel3

"Definir Variable" -> Nivel2

"Obtener Variable" -> Nivel1

"Ajustar Valor del Diccionario"-> "infotainment"
se actualiza a la variable Nivel2

"Definir Variable" -> Nivel1

"Obtener Variable" -> Diccio

"Ajustar Valor del Diccionario"-> "extras" se actualiza a la variable Nivel1

"Definir Variable" -> Diccio

Finalmente se guarda el archivo en la nube sobrescribiendo el JSON inicial.

Ejemplo descargable desde:



[Descárgalo aquí](#)

Como has podido observar, este último ejemplo es más laborioso pero no deja de utilizar la misma secuencia que el ejemplo 1, eso sí, con más complejidad debido a los sub niveles.

Con esta última parte se cierra la parte de diccionarios, sin duda un aspecto importante para los shortcuts de nivel más avanzado.

7.5 Estructuras de control

Las **estructuras de control** son expresiones que evalúan y modifican el orden de ejecución de las instrucciones dentro de un programa. A menos que el flujo del programa se vea modificado por una expresión de control, las instrucciones siempre se ejecutan secuencialmente, una detrás de otra.

En Shortcuts las acciones se ejecutan en orden secuencial, de arriba a abajo. Conforme crees shortcuts más complejos necesitarás utilizar acciones que modifiquen el orden de ejecución de este flujo.

7.5.1 Si (también conocido como “If”)

Las acciones condicionales **Si** se utilizan para configurar diferentes flujos de ejecución dentro de un shortcut. Estas evalúan una operación lógica que da como resultado *verdadero* o *falso*, y ejecuta un bloque de acciones siempre y cuando el resultado sea *verdadero*. Dicho de otra forma: *si una condición es verdadera, haz eso; si no, haz lo otro*.

Estas acciones tienen dos salidas (verdadero o falso), cuando se ejecuta el shortcut y se evalúa la condición, una respuesta verdadera envía el flujo hacia la primera acción justo después de **Si**; mientras que las condiciones falsas lo dirigen a la primera acción después de **Si no**.

La evaluación de si es verdadero o falso para tomar el camino adecuado se hace comparando la entrada al **Si** con un valor. Dentro de la propia acción **Si**, bajo el parámetro “entrada” tienes la opción de comparar de cuatro modos:

- Es igual a
- Contiene
- Es mayor que
- Es menor que

Si comparas un objeto de tipo texto solo podrás usar las dos primeras opciones, ya que el mayor y el menor están destinados a comparar números.

Para el texto, la coincidencia tiene que ser exacta, es decir, si la entrada es la palabra “Casa” y en la acción **Si** pones “Casa” dará verdadero. Sin embargo, al evaluar con el valor “casa” con minúscula, será falsa. Igual ocurre con las tildes y otros

caracteres. Por tanto debes poner atención al usar el texto como entrada del **Si**.

En el caso de variables numéricas, puedes usar cualquiera de las cuatro opciones.

En los comparativos faltan dos opciones que se usan a menudo en otros entornos de programación: “menor o igual que” y “mayor o igual que”. Hay una manera de suplir esta carencia, con un ejemplo lo entenderás mejor:

Ejemplo 1. Necesitas un **Si** que se ejecute solo si el valor es mayor o igual que 2. La configuración del **Si** será:

- Entrada: es mayor que
- Valor: 1

Al seleccionar solo los números mayores que 1, el primer valor coincidente es desde el 2 en adelante.

En caso de que necesites que dé *verdadero* solo si la entrada es menor o igual a 2 la configuración será esta:

- Entrada: es menor que
- Valor: 3

El primer número menor que 3 es el 2, por lo que la condición si, se cumplirá con el 2, el 1 y el 0 y todos los números negativos.

Usa [este shortcut](#) para comprobar los ejemplos y entender mejor cómo funciona el Si.



[DESCÁRGALO AQUÍ](#)

7.5.2 Repetir (For)

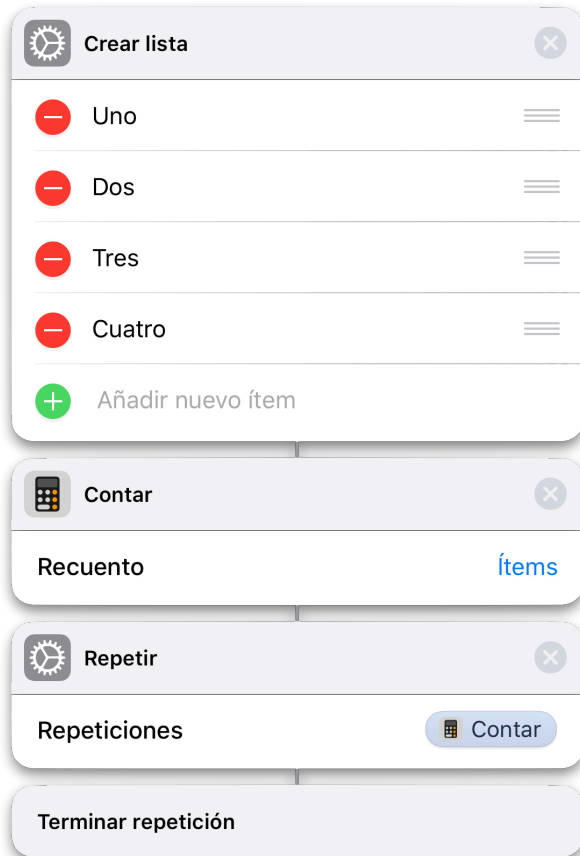
Las acciones **Repetir** funcionan de la misma manera que los bucles **for**, son estructuras de control que permiten indicar el número de veces que se debe repetir un conjunto de acciones.

En Shortcuts, al ejecutarse una acción **Repetir**, el grupo de acciones que contiene en su interior se ejecutan en orden secuencial; cuando llega a la última, el flujo vuelve a la primera acción y se ejecuta de nuevo, así tantas veces como se le indique.

Con cada repetición, la salida de la última acción se coloca en una lista, esta lista será la salida de la acción **Repetir**.

Para asignar el número de repeticiones al valor de una variable pulsa el espacio vacío a la izquierda de los símbolos (+) y (-).

Con frecuencia necesitarás que un bloque de acciones se repita un número de veces concreto que no corresponda al número de elementos de la lista de entrada, esto se explicará en el siguiente punto. En ese caso puedes usar la acción “contar” para obtener el número de elementos de una lista, el número de palabras de un texto... Puedes introducir la salida de la acción contar en el número de repeticiones.



CONTAR ELEMENTOS PARA CONFIGURAR EL NÚMERO DE REPETICIONES

7.5.3 Repetir con cada uno (For Each)

La acción **Repetir con cada uno** es una variación de la anterior en la que el número de repeticiones es igual al número de elementos o ítems que posea el objeto de entrada. Se utiliza para recorrer estructuras repetitivas.

Como ves, “Repetir con cada uno”, es un “Repetir”, en el que está preconfigurado el contar los elementos de la lista.

7.5.4 Seleccionar del menú (Select-Case)

Cómo se puede deducir por su nombre, la acción **Seleccionar del menú**, muestra un menú con varias opciones y ejecuta el bloque de acciones asignado a la opción seleccionada. Se utiliza para agilizar la toma de decisiones entre múltiples posibilidades.

Por cada acción **Seleccionar del menú** sólo se puede ejecutar una de las opciones. Si quieres que al finalizar una de las opciones del menú este vuelva a aparecer ofreciéndote las opciones puedes “simular” este comportamiento con la acción “Ejecutar atajo” y selecciona el mismo shortcut, así volverá a ejecutarse desde el principio.

8. Web APIs

Shortcuts se ha convertido en una gran herramienta porque no se ha visto encajada con sus bloques predefinidos, sino que gracias a los x-callback-url y los URL Schemes (véase capítulo 3) ha podido “expandir” su utilidad. Su antecesora, WorkFlow, en su actualización [1.5.3](#), marcó un antes y un después, se incluyeron un conjunto de Acciones que permiten al usuario el uso de Web APIs.

Antes de entrar en una de las partes más densas del libro, vas a aprender en forma de analogía lo que es una API. Imagina una biblioteca, donde puedes pedir un libro, los libros de un autor específico, los libros de una editorial, etc... La biblioteca sería la API, y cada uno de los modos en los que puedes hacer una consulta es una “*request*” (petición). Además, ahora piensa que para hacer cada petición debes demostrar al empleado qué eres abonado y que tienes el carnet de socio. Esto es básicamente una API. ¿Qué se entiende como “Web API”? Que la petición sigue una estructura y un protocolo determinado. Siguiendo el ejemplo, imagina que para hacer cualquier petición (autenticar que eres tú, y cualquier consulta) debieras enviar la petición mediante una carta de color rosa los lunes por la tarde.

Shortcuts ha traspasado sus propias limitaciones con las acciones predefinidas, y ahora somos capaces, sin entrar en detalles de librerías, código, etc... de llamar a servicios web / Web APIs que están online y que retornarán información para poder trabajar con ella, haciendo de Shortcuts una completa navaja suiza.

Con ese “superpoder”, Shortcuts ya no dispone solo de sus propios bloques que interactúan con los servicios nativos del sistema (calendario, música...), de las aplicaciones de terceros instaladas en tu iPhone/iPad (Fantastical, Bear,...), Sino que ahora es capaz de “*hablar*” con la Web. Hay un

mundo de posibilidades que se abren en Shortcuts gracias a dos tipos de “acceso” a la Web: Web Api y servicios Web.

Veamos en un lenguaje más formal lo que es una API.

8.1. Qué es una API

Una API (*Application Programming Interface*) es la interfaz que una aplicación utiliza para comunicarse con otra. Estas interfaces permiten usar funcionalidades de una forma simple sin saber realmente lo que está sucediendo en el otro lado.

Las API en general facilitan el trabajo a los desarrolladores ya que se aprovecha lo que en programación se llama [capa de abstracción](#). De esta manera, el desarrollador no tiene porqué saber lo que ocurre dentro de esa caja llamada API, y lo único que debe es saber a qué función llamar y cómo y de qué forma pasar los parámetros necesarios. Una vez dentro de esa “caja negra” se realizan las operaciones necesarias de forma transparente y el resultado es devuelto. Dicho de otra manera, una API permite implementar las funciones y procedimientos que engloba en nuestro proyecto sin la necesidad de programarlas de nuevo.

Todas tus apps favoritas funcionan haciendo llamadas a una API, Dropbox, Facebook, Pocket... Si las APIs son públicas, gracias a Shortcuts se podrá disponer de acceso a ellas y por lo tanto nos situamos en un plano totalmente diferente al de limitarnos a los bloques que la aplicación dispone de serie.

Se ha dicho API pública. La mayoría de servicios que brindan su acceso a hacer estas llamadas te obligarán a menudo a obtener primero un acceso. El motivo es fácil de intuir, y es que quieren que se haga un uso racional de ellas para evitar el colapso en sus servidores. El acceso más habitual puede ser vía user/password o vía API Token. Tanto para uno como para lo otro necesitarás siempre registrarte previamente en el

servicio a “explotar”. Todo esto se explica más adelante, así que no te preocupes por ahora.

Muchos sitios y aplicaciones Web exponen parte de sus funcionalidades a través de APIs, permitiendo así extender su funcionalidad de maneras que ni los propios creadores de la API habrían imaginado.

Habrás observado que hay más de una app para conectar a Twitter, esto es porque su API es pública y solo debes darte de alta como desarrollador en su servicio para tener acceso a dichas APIs. Sin embargo solo hay una aplicación para conectar a Facebook, en este caso la API es cerrada y solo Facebook tiene acceso a ella.

8.2. Web Services: SOAP APIs y REST APIs

Antes de avanzar más será necesario ampliar un poco de conocimientos. Para ello, vas a aprender que es un Web Service (Servicio Web).

Todas las APIs que usan el protocolo HTTP como formato para el transporte de las peticiones/respuestas se pueden clasificar como Web Services. Un Web service es una aplicación web que provee *recursos* en un formato accesible para otros clientes. Los Web Services incluyen diferentes tipos de API, incluyendo **REST APIs** y **SOAP APIS**. En resumen, básicamente los Web Services se basan en peticiones y respuestas entre clientes y servidores.

Con los Servicios Web, el cliente que hace la petición para un recurso y la API que corre en el servidor pueden usar distintos lenguajes de programación o plataforma. Esto no importa porque la respuesta será vía protocolo HTTP.

Esta es una de las grandezas de los Servicios Web, son independientes del lenguaje, son interoperables a través de distintas plataformas y sistemas. Cuando se documenta una

REST API por ejemplo, no importa que la API esté en JAVA, Ruby, Python... Las peticiones son hechas vía protocolo HTTP, y las respuestas, de igual manera.

Eso si, cada lenguaje tendrá su propio modo para realizar esa consulta, pero la respuesta de la API no es una página HTML, sino que a menudo la respuesta es en forma de Fichero [JSON](#), [XML](#), CSV... pero JSON es a día de hoy el más común. (Se explicó JSON en el capítulo dedicado para ello (5.5.)

Los Servicios Web que se han comentado anteriormente incluían SOAP APIS y REST APIS. Se va a explicar un poco que son, pero como avance, decirte que las que te interesan son las REST API (las más habituales actualmente)

SOAP API

Es un protocolo que requiere de XML como formato para las peticiones y respuestas. Como todo estándar, significa que todo mensaje deberá seguir unas normas de formato, en el caso de SOAP se llama WSDL (Web Service Description Language). Lo que significa todo esto es que WSDL define los elementos y atributos (para saber más, [XML](#)

¿Por qué SOAP perdió terreno ante **REST API**? El principal problema de SOAP es que el formato de los mensajes es muy recurrente y pesado. Esta circunstancia provoca que en entornos de movilidad no sea el escenario ideal para facilitar las comunicaciones, satura el ancho de banda y provoca lentitud. A día de hoy hay servicios basados en SOAP, pero sobre todo en aplicaciones de tipo empresarial.

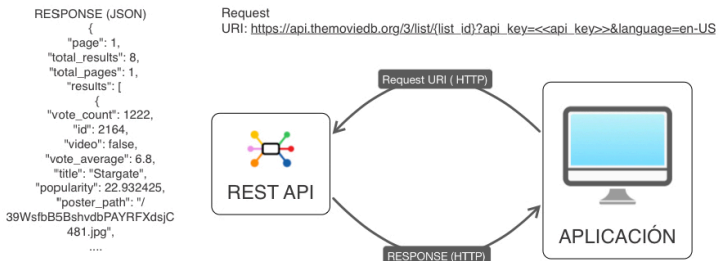
REST APIS

Como SOAP, REST usa HTTP como protocolo de transporte para peticiones y respuestas. Sin embargo, a diferencia de SOAP, NO es un protocolo standard.

Una **REST API** puede no seguir las bases definidas por el [Dr. Roy Fielding](#), que fue quien primero describió el modelo.

El modelo de una REST API se puede representar de este modo :

MODELO DE UNA REST API



REST API

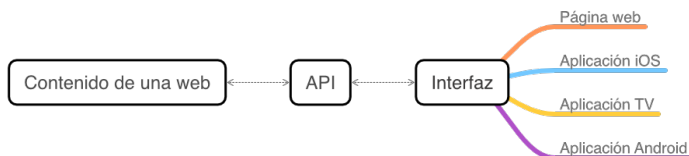
Como puedes ver, hay una petición y una respuesta entre un cliente y el API server. Como se ha comentado anteriormente, el cliente y el servidor son independientes del lenguaje en que trabajen, el protocolo HTTP se encarga de todo. En el caso del ejemplo, la respuesta es en formato JSON, pero podía ser XML, JSON, CSV, HTML... Estos tipos de fichero contienen la información que devuelve la API a la petición que hemos realizado.

A pesar de todas las posibilidades, JSON es sin duda alguna el formato más habitual usado en las REST APIs. ¿El motivo? Es sencillo, liviano y flexible. En escenarios de movilidad ocupa poco ancho de banda, y como verás, es muy fácil de “parsear” (leer su estructura). Shortcuts es capaz, usando el bloque “*Obtener Diccionario de la entrada*”, de tener como entrada a la acción un fichero JSON y a partir de ese punto,

de una manera fácil, recuperar cualquier valor o incluso recorrerlo con un “*Repetir con Cada Uno*”.

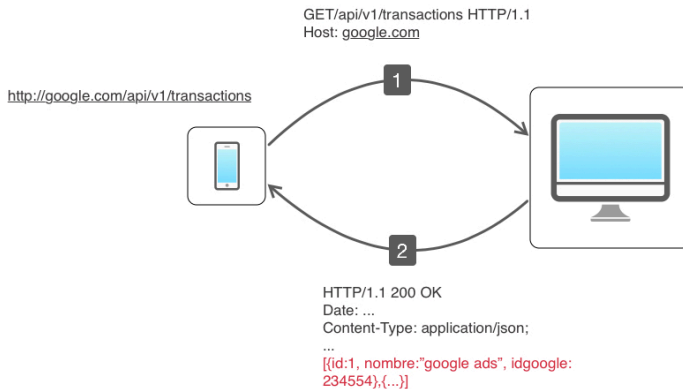
Cualquier web moderna se estructura mediante un sistema que permita la mayor interoperabilidad posible, ya que los contenidos de la web serán consumidos desde un navegador y posiblemente desde una app para iOS, otra para Android, Windows Mobile, en algunos casos una TV,...

Es por eso que se separa el contenido de la web de la forma en que se muestran (webs y apps). Y la forma en que se pasa información entre ambos núcleos es la Web API. Observa el gráfico.



ESQUEMA BÁSICO DE UNA WEB USANDO API

Esto permite al desarrollador responder de forma rápida ante un cambio o una actualización, y que haya equipos distintos trabajando en el núcleo de la web y otros en cada una de las interfaces. No te preocupes si todo está un poco confuso poco a poco irás viendo la diferencia entre todos estos conceptos.



WEB API

8.3. Recursos

Otro aspecto de las REST APIs es que se centran en los recursos (datos devueltos por la API) y el modo de acceder a ellos. Como has podido ver, se accede a los recursos a través de URLs (*Uniform Resource Locators*). Éstas van acompañadas con los llamados “métodos”, y que especifican como se quiere interactuar con este recurso. Los métodos más comunes son GET (leer) , POST (crear), PUT (actualizar) y DELETE (borrar), lo que se conoce como verbos.

De acuerdo, respira. No debería ser una prioridad entender todo lo escrito, pero si se ha creído necesario dar una visión general de lo que es una API, y más específicamente una REST API. No es objetivo del libro dar una master class, pero para poder utilizarlas con Shortcuts deberás saber un poco más.

La relación entre los recursos (datos) y los métodos se puede describir como Nombres y Verbos. El recurso es el nombre

porqué es un objeto, el verbo es lo que quieres hacer con el nombre (DELETE,POST,GET...). Combinando nombres y verbos es como se da forma a REST.

En cualquier documentación de una REST API verás algo parecido a esto:

```
GET /campaigns
GET /campaigns/{campaign_id}
POST /campaigns/{campaign_id}
DELETE /campaigns/{campaign_id}
PATCH /campaigns/{campaign_id}
```

Son los llamados **EndPoints**. Como ves, se cita primero el verbo a usar (GET,POST,PATCH,DELETE), y una especie de URL (normalmente no se escribe completa, por eso la ves extraña).

Los verbos: Te indican que acción va a ocurrir.

- GET: Recupera un recurso(datos)
- POST: Crea un recurso
- PUT: Actualiza o crea un recurso
- PATCH: Modifica parcialmente un recurso existente.
- DELETE: Elimina un recurso

Los **endpoints** indican como acceder al recurso, es decir, la URL final, y el método usado en el endpoint indica el tipo de acción (GET,POST DELETE) a usar con el recurso. Explicado en lenguaje llano, un endpoint es una URL que en vez de devolver una página, nos devolverá un JSON (CSV, XML...).

Un mismo recurso a menudo tiene distintos endpoints, cada uno con rutas y métodos que devuelven diferente información según proceda. Por ejemplo, el GET está definido para leer información por lo que devolverá un JSON con datos, y POST para “subir información”, por lo que devolverá un JSON indicando si todo ha ido bien, etc..

Un **endpoint** normalmente incluye algún parámetro que especifica más detalles al respecto de lo que queremos hacer. Los parámetros van entre “{}”.

Ejemplos de Endpoints de REST APIs:

```
POST https://api.spotify.com/v1/users/  
{user_id}/playlists  
GET https://api.spotify.com/v1/me/player/  
recently-played
```

Normalmente una API dispondrá de un número de endpoints agrupados por recurso. En el ejemplo siguiente, el recurso “clients” tiene varios endpoints tal y como quedan descritos :

```
POST https://www.toggl.com/api/v8/clients  
GET https://www.toggl.com/api/v8/clients/  
{client_id}  
PUT https://www.toggl.com/api/v8/clients/  
{client_id}  
DELETE https://www.toggl.com/api/v8/clients/  
{client_id}
```

Aquí puedes ver el verbo (POST, GET..), el recurso (clients) y los parámetros (Client_id). Los parámetros son opciones que puedes pasar a un endpoint, y son representados, como se ha apuntado en el párrafo anterior, mediante “{}”.

En las documentaciones sobre APIs a menudo verás solo la parte final de la URL, de ahí el término endpoint. Resumiendo:

Cuando se describe un endpoint, normalmente no se presenta toda la ruta (en este caso si que está toda para que quedara más claro). A la ruta completa se la llama comúnmente **Recurso URL**.

(Base de la URL) + EndPoint = URL_del recurso.

Siguiendo el ejemplo de la API de [Tooggl](#):

/clients/{client_id} es el **endpoint**

https://www.toggl.com/api/v8 es el **base path** (ruta base)

Se debe tener en cuenta que la documentación de cada API variará, es decir, no hay un patrón o guía, habrá quien agrupe endpoints por verbo (método), o por el contrario, separar la documentación por recursos, etc...

Las APIs no son lectura fácil y a menudo implican alguna que otra prueba, por eso es útil tener algunas armas (conocimiento básico) de como está formada una REST API.

En el ejemplo de la API anterior (es la API de [Toggl](#)), has podido ver que el recurso "client" estaba entre "{}". Como he apuntado anteriormente, se llaman parámetros.

Los parámetros no son de carácter obligatorio y dependerá en gran medida del recurso y del verbo que queramos utilizar. Hay 4 tipos de parámetros: HEADER, PATH, QUERY STRING y REQUEST BODY.

Observa notar el carácter opcional, es decir, los endpoints dispondrán de alguno, todos o ningún parámetro según esté implementado.

- **HEADER:** Parámetros que están incluidos en el encabezamiento de la petición. Normalmente son para autorización.
- **PATH :** Parámetros que aparecen entre la ruta del endpoint, antes de los parámetros ("query String")
- **Query String :** Parámetros que aparecen después del "?".
- **Request Body :** Parámetros incluidos en forma de clave-valor con formato JSON.

En una REST API es igual importante saber los parámetros a pasar como el formato de estos parámetros. Los diferentes formatos son los estándar: String (alfanumérico), integer

(número), Boolean(verdadero/Falso), Array (lista de valores), Object (clave - Valor en formato JSON). Puede haber más, pero me quedo en los soportados por WorkFlow.

8.4. Otros Aspectos de las REST APIs

Para ir terminando con este tema tan denso y antes de relacionarlo con Shortcuts, un par de apuntes más sobre las REST APIs.

En toda documentación, verás que se hace referencia a la respuesta del protocolo HTTP de como ha ido la transacción, y de como ha respondido el servidor. Dependiendo de la ejecución de la consulta, el servidor devolverá un código proporcionando información del estado de la comunicación.

HTTP 100–199 Informativas

HTTP 200 299 OK

HTTP 300–399 Redirecciones

HTTP 400–499 No correcto

HTTP 500–599 Error interno del servidor

Mira un ejemplo de los códigos de estado de la API de [TheMoviedb](#) :

Example Response

```
{
  "status_code": 7,
  "status_message": "Invalid API key: You must be granted a valid key."
}
```

Code	HTTP Status	Message
1	200	Success.
2	501	Invalid service: this service does not exist.
3	401	Authentication failed: You do not have permissions to access the service.
4	405	Invalid format: This service doesn't exist in that format.
5	422	Invalid parameters: Your request parameters are incorrect.
6	404	Invalid id: The pre-requisite id is invalid or not found.
7	401	Invalid API key: You must be granted a valid key.
8	403	Duplicate entry: The data you tried to submit already exists.
9	503	Service offline: This service is temporarily offline, try again later.
10	401	Suspended API key: Access to your account has been suspended, contact TMDB.
11	500	Internal error: Something went wrong, contact TMDB.
12	201	The item/record was updated successfully.
13	200	The item/record was deleted successfully.
14	401	Authentication failed.
15	500	Failed.
16	401	Device denied.
17	401	Session denied.
18	400	Validation failed.
19	406	Invalid accept header.
20	422	Invalid date range: Should be a range no longer than 14 days.
21	200	Entry not found: The item you are trying to edit cannot be found.
22	400	Invalid page: Pages start at 1 and max at 1000. They are expected to be an integer.
23	400	Invalid date: Format needs to be YYYY-MM-DD.
24	504	Your request to the backend server timed out. Try again.
25	429	Your request count (#) is over the allowed limit of (40).
26	400	You must provide a username and password.
27	400	Too many append to response objects: The maximum number of remote calls is 20.
28	400	Invalid timezone: Please consult the documentation for a valid timezone.
29	400	You must confirm this action: Please provide a confirm=true parameter.
30	401	Invalid username and/or password: You did not provide a valid login.
31	401	Account disabled: Your account is no longer active. Contact TMDB if this is an error.
32	401	Email not verified: Your email address has not been verified.
33	401	Invalid request token: The request token is either expired or invalid.
34	401	The resource you requested could not be found.

CÓDIGOS DE RESPUESTA

8.4.1. SEGURIDAD EN LAS REST APIS

Se va a acabar esta parte hablando de seguridad. En la casi totalidad de las APIs se pide autenticación mediante varios métodos para saber quién y cómo está usando esa API.

Las consecuencias de no asegurar la autenticación (proveer la identidad correcta) y Autorización (Permitir una determinada acción) en las peticiones API pueden ser varias:

- Los usuarios podrían ejecutar llamadas ilimitadas a la API, provocando saturación de servidores.
- No habría control, nadie sabría cuantas llamadas ha hecho un usuario, etc..

- Se podría interceptar información y no habría manera de saber quién fue.

Por lo tanto, como podrás comprobar (sino lo has visto ya), la mayoría de APIS requieren de una autorización para poder usar la API. Esto implica que controlan quien hace cada petición, pueden bloquear a un usuario que exceda el número de peticiones, e incluso aplicar diferentes roles según el usuario.

Las REST API tienen diversas formas para dotar de seguridad a la API :

1. Esquemas de autenticación HTTP
 - BASIC
 - Bearer
 - Otros
2. API KEYS
 - En el HEADER
 - En forma de QUERY STRING
 - Cookies
3. OAuth2
4. OpenID Connect Discovery

Se explicarán los que afectan a Shortcuts y que según la API que quieras atacar encontrarás en su documentación, no es objeto del autor entrar en temas que se alejan ya del objetivo, es decir, poder atacar REST APIs desde Shortcuts:

Basic Auth : Este método es el más sencillo, el cliente envía `username:password` en el HEADER. Ambos están codificados en **Base64**. De este tipo de codificación se habla en el capítulo 5.4.2. Un ejemplo:

`Authorization: Basic lY3VQ=yZbG=9s0nN`

Las APIS que usan Basic Auth deberían usar HTTPS, así se asegura que el mensaje estará encriptado dentro del protocolo de transporte HTTP. Sin HTTPS es muy fácil decodificar un

mensaje en Base64. Destacar la diferencia entre CODIFICADO y ENCRIPTADO, no es lo mismo.

Un ejemplo lo puedes ver en la documentación de [Tinypng](#), una [API](#) para tratar imágenes:

Authentication

To use the API you must provide your API key. You can [get an API key](#) by registering with your name and email address. Always keep your API key secret!

Authentication to the API is done with [HTTP Basic Auth](#). All requests require an `Authorization` header that contains a Base64 digest of the authentication string `api:YOUR_API_KEY` where `YOUR_API_KEY` is the key that can be found on your API account page.

All requests must be made over an encrypted [HTTPS](#) connection.

CAPTURA DE LA DOCUMENTACIÓN DE LA API DE TINYPNG

Su uso en Shortcuts:

Texto

api: Texto

Codificar con Base64

Modo Codificar Descodificar

Salto de línea Cada 64 caracteres

URL

URL <https://api.tinify.com/shrink>

Obtener contenido de URL

Avanzado ∨

Método POST

Cabeceras ∨

− Authorization Basic Codificación Ba ≡

+ Añadir nueva cabecera

Cuerpo de la solicitud Archivo

Archivo Tamaño de imagen modificado

EJEMPLO BASIC AUTH



DESCÁRGALO AQUÍ

API KEYS: Algunas APIS no usan usuario y password, y lo que hacen es generar lo que se llama una API KEY (método muy común). Esta API key hay dos maneras de enviarla :

- En la QUERY STRING :

```
GET /url?api_key=TU_API_KEY:Por ejemplo  
la API de Google:
```

```
https://maps.googleapis.com/maps/api/  
directions/json?  
origin=Toronto&destination=Montreal&key=YOU  
R_API_KEY
```

- En el HEADER (encabezado):

```
GET /something HTTP/1.1 Ocp-Apim-  
Subscription-Key: abcdef12345:
```

Por ejemplo, Microsoft Recognition Services :

```
POST "https://  
westus.api.cognitive.microsoft.com/vision/  
v1.0/recognizeText?handwriting=true"  
-H "Content-Type: application/json"  
-H "Ocp-Apim-Subscription-Key:  
{subscription key}"
```

Bearer Authentication: También llamada “Token Authentication”. Se puede entender como “concede permiso al poseedor de este token”.

El “Bearer Token” es un string encriptado, normalmente generado por un servidor en respuesta a una petición de login.

El cliente pasa el Token obtenido en el HEADER cuando hace peticiones a datos más “sensibles” o de carácter más privado.

Este tipo de autenticación fue creada primero dentro de Oauth 2.0, pero se puede implementar de forma autónoma. Puedes ver un ejemplo en la API de [CloudConvert.com](https://cloudconvert.com):

API de CloudConvert

Creating a new Process ID

In order to start a new conversion you need to create a new Process ID first. Each Process ID is for one-time use only and can only be used for converting one file.

Raw Request cURL PHP node.js Python

Example Request

```
POST https://api.cloudconvert.com/process
```

Authorization: Bearer API_KEY

```
{
  "inputformat": "flv",
  "outputformat": "mp4"
}
```

Parameter	Description
<code>inputformat</code>	* Current format of the file. (Required)
<code>outputformat</code>	* Output format, to which the file should be converted. (Required)
<code>mode</code>	The mode of the process. Defaults to <code>convert</code> . Other possible values are <code>info</code> , <code>combine</code> , <code>archive</code> and <code>extract</code> . (Optional)

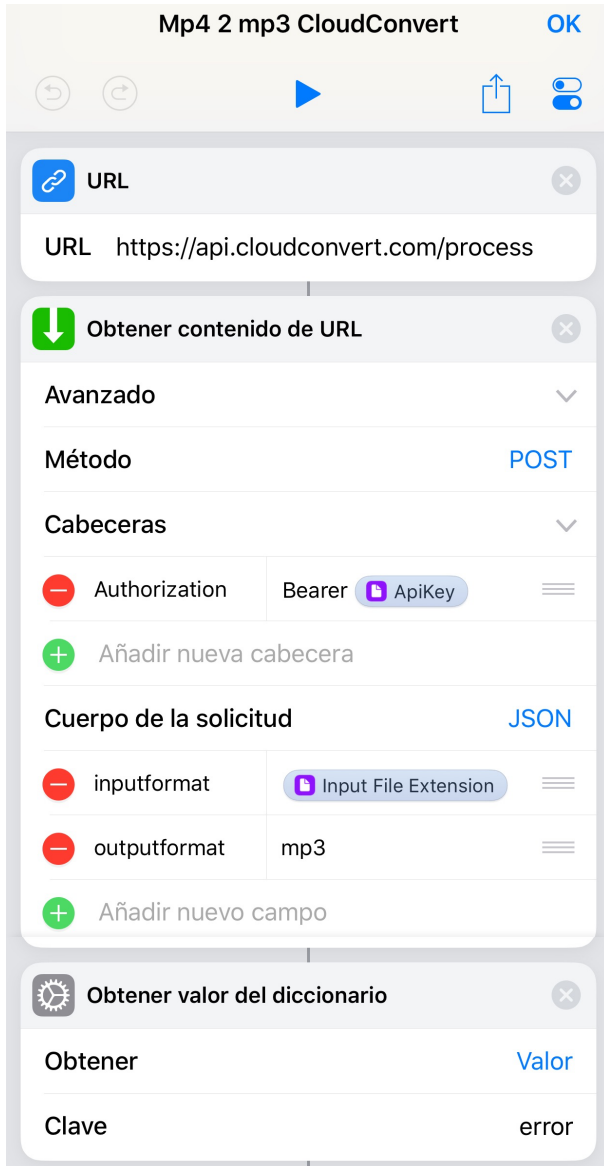
A response could look like this:

Example Response

```
{
  "url": "//srv01.cloudconvert.com/process/v4cw72hf3",
  "id": "v4cw72hf3",
  "host": "srv01.cloudconvert.com",
  "expires": "2014-09-12 13:13:00",
  "maxtime": 18000,
  "minutes": 264
}
```

PETICIÓN DE UN TOKEN

En un shortcut:



EJEMPLO DE PETICIÓN QUE INCLUYE AUTENTICACIÓN DENTRO DE LA PETICIÓN USANDO BEARER.



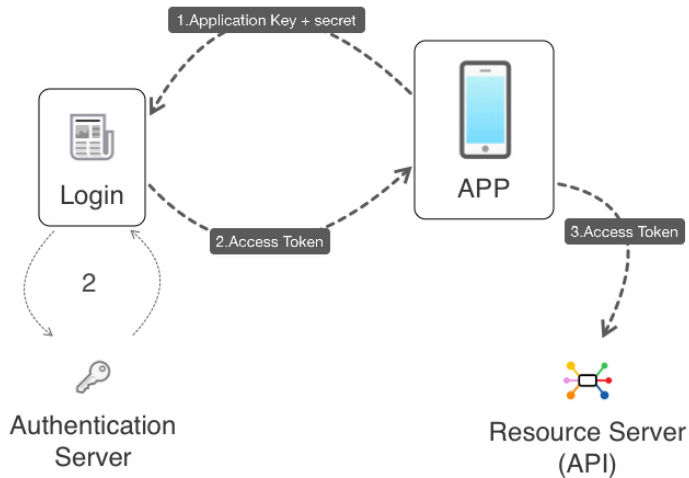
[DESCÁRGALO AQUÍ](#)

OAuth 2.0 (Bearer Token Scheme): Es un protocolo de autorización que otorga permisos limitados de un usuario a datos en un servidor. Github, Google, Twitter, Pocket... son ejemplos claros. Es decir, permite a terceros (cliente) acceder a contenidos propiedad de un usuario (alojados en aplicaciones de confianza, servidor de recursos) sin que estos tengan que conocer las credenciales del usuario. De esta forma estas aplicaciones de terceros pueden acceder a datos sensibles del usuario, pero no son concedoras de las credenciales de autenticación.

Hay varios tipos de Oauth, pero lo importante a saber es que **Shortcuts no soporta Oauth 2.0**, ya que este sistema requiere al usuario introducir manualmente su user y password en una página de login, y esta devuelve un token que es el que usará para acceder a la API.

El funcionamiento básico de Oauth 2.0 es el siguiente :

OAuth2.0



ESQUEMA DE FUNCIONAMIENTO DE OAUTH2

1. El cliente solicita autorización al propietario del recurso (usuario). Esto habitualmente es una página de login donde nos pide credenciales. Si el usuario autoriza la petición, la app recibe acceso a pedir autorización.
2. La aplicación seguidamente hará una petición al Servidor de Autenticación. Si es exitosa, devolverá un Token de acceso.
3. Finalmente, la aplicación hará una petición al servidor de Recursos (donde están los datos, el API SERVER), pasando el Token de acceso. El servidor devolverá los datos.

OAuth2 tiene dos ventajas :

1. Soluciona el problema de confianza entre usuario y aplicaciones de terceros.

2. Permite a un proveedor de servicios/API facilitar a las apps de terceros ampliar sus funcionalidades dando el acceso a datos sensibles de manera segura. El usuario podrá en cualquier momento permitir o denegar el acceso a sus datos teniendo control de quien puede acceder a sus datos más sensibles.


Por ejemplo, [Pocket](#) tiene en su configuración la posibilidad de permitir/denegar acceso a las apps que hayamos autorizado :

Aplicaciones y servicios conectados

Has concedido acceso a tu cuenta de Pocket a las siguientes aplicaciones. Puedes anular el acceso haciendo clic en el enlace que hay junto a cada aplicación.

Servicios conectados








Has autorizado a Pocket para que tenga acceso a las siguientes cuentas:

-  Twitter - [Anular acceso](#)

Aplicaciones conectadas

Has conectado las siguientes aplicaciones y extensiones:

Aplicaciones oficiales de Pocket

-  Pocket for Android - [Anular acceso](#)
-  Pocket for Chrome Packaged App - [Anular acceso](#)
-  Pocket for Google Chrome - [Anular acceso](#)
-  Pocket for iPad - [Anular acceso](#)
-  Pocket for iPhone - [Anular acceso](#)
-  Pocket for Mac - [Anular acceso](#)
-  Pocket for Safari - [Anular acceso](#)

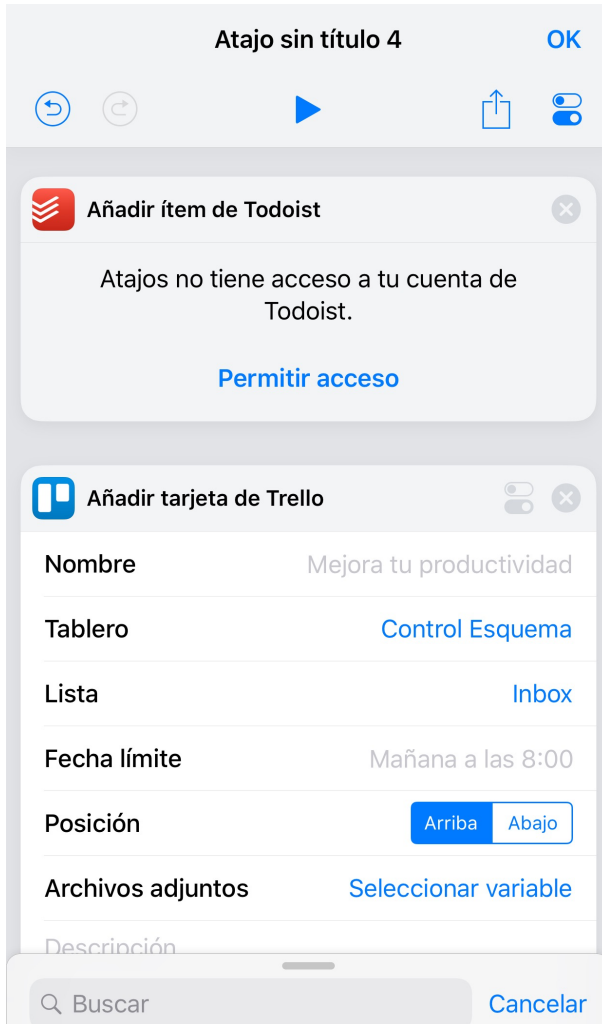
PANTALLA DE CONFIGURACIÓN DE ACCESO DE OAUTH2 EN POCKET

8.5. Resumen y Ejemplos

En este punto ya deberías tener claro que Shortcuts, además de tener acciones nativas que lanzan aplicaciones, gracias a los URL Schemes, puede ir más allá y “hablar” con servicios Web. Por lo tanto, se pueden realizar determinadas acciones sin la necesidad de tener una app instalada para ello.

Después de toda la teoría anterior, y antes de entrar en ejemplos, para aclarar nuevamente conceptos, hay que tener claro que hay disponibles dos tipos de acciones Web en Shortcuts :

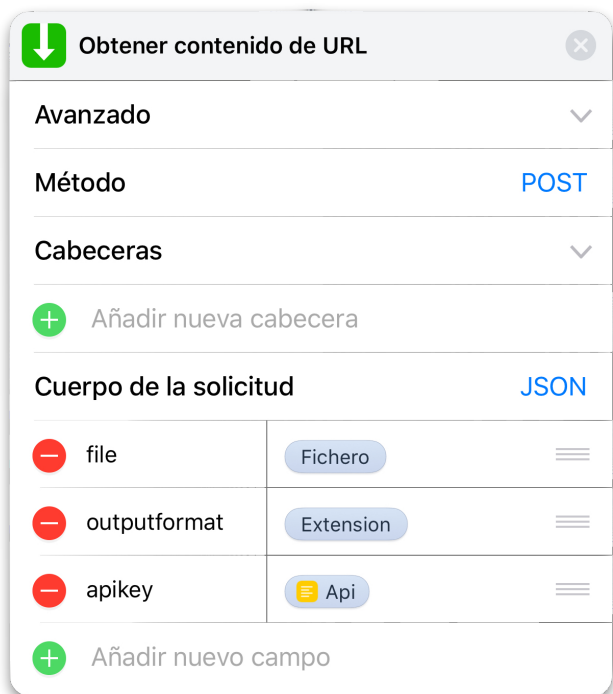
- **Acciones predefinidas:** Son las típicas con una interfaz gráfica representada en forma de un bloque más. Este tipo requiere un mantenimiento de los desarrolladores de Shortcuts para que el servicio Web sea “compatible” y podamos usarlo bajo una interfaz gráfica muy sencilla para los usuarios. La integración con estos servicios ya está realizada toda “por detrás” de Shortcuts, y tú solo necesitarás dar “permiso” a la aplicación para acceder a ese servicio. Ejemplos de ello son Trello, ToDoist, WordPress, Pocket, Dropbox... Hay dos “especiales” que no requieren autenticación que son iTunes y App Store, que están disponibles para usar sin ningún tipo de validación de usuario.



DOS APLICACIONES PREDEFINIDAS, UNA SOLICITANDO ACCESO OTRA, CON EL ACCESO CONCEDIDO PREPARADA PARA INTRODUCIR LOS PARÁMETROS.

- **“Obtener Contenido URL”**: Este segundo modo es un modo más “directo” y requiere de mas trabajo, ya que es

el propio usuario quien debe “construir” la comunicación con el servicio Web, es decir, construir la integración con la API de ese servicio.



UNA ACCIÓN DE “OBTENER CONTENIDO DE URL” CONFIGURADA PARA ACCEDER A UNA API.

Un caso que ejemplifica claramente la anterior clasificación es la producida en la versión 2.1 de [Shortcuts](#) (30 Oct. 2018), en la que se incorporan dos bloques referentes a la previsión meteorológica. Anteriormente para poder obtener estos datos se debía realizar una petición via API a servicios como [Dark Sky](#), con toda la dificultad que implica y el freno que supone para usuarios standard, aunque una hayas terminado y practicado este capítulo deberías poder hacerlo por tus propios medios.

Otros ejemplos anteriores fueron [Pocket](#) que, en su momento tuvo bloques en Workflow y tras la adquisición de Workflow por parte de Apple estas acciones se retiraron para volver a aparecer más tarde.

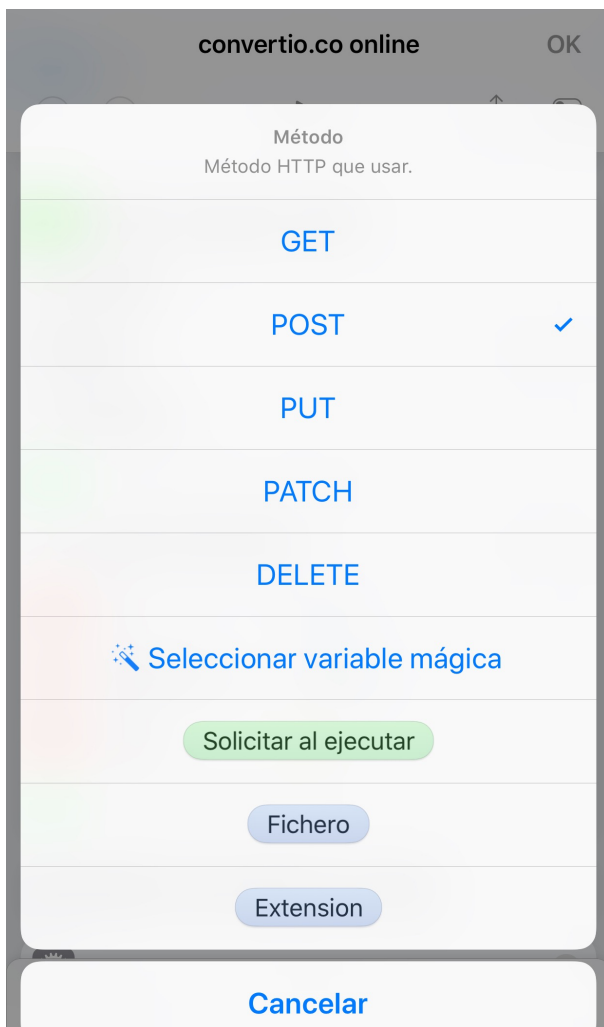
Como se ha estado explicando a lo largo de los diferentes apartados Shortcuts es capaz de usar APIs, más concretamente REST API. Al final, toda la teoría anterior ha servido para llegar a este punto. Lógicamente no es vital tener conocimiento al 100% de todo lo explicado en este capítulo pero con seguridad que te ahorrará muchas pruebas ensayo-error si has entendido los conceptos. En el peor de los casos, tienes en este libro un lugar de consulta donde poder apoyarte.

De todos los verbos nombrados en el punto 8.3., en muchas ocasiones usarás GET para obtener datos, y POST para subirlos y hacer una petición al servidor. Estos dos verbos son los más usados sin duda. El cómo se realizan estas llamadas dependerá de cada API. Sí, esto implica ir a la documentación y leer para cada caso particular como debe llamarse a cada uno de los métodos de la API.

El bloque “*Obtener Contenido URL*” tiene un funcionamiento muy básico y otro mucho más complejo que aplica todo lo explicado anteriormente. En el modo simple, se le pasa una URL como entrada y devuelve el contenido de esa URL. En el modo avanzado, se puede ver un desplegable que permite configurar opciones y aquí es donde se abre un mundo de posibilidades.

A pesar de todo este conocimiento adquirido te espera alguna que otra pelea con estas configuraciones, principalmente por el desconocimiento o la dificultad de entender la documentación de alguna API (o incluso documentación errónea). Por lo tanto, un consejo, paciencia, perseverancia y releer lo anterior hasta tener claro lo que es cada actor en una documentación de una REST API.

Cuando uses esta acción en modo avanzado, la primera opción que deberás escoger es el **verbo** http (GET,POST,PUT,DELETE,PATCH)

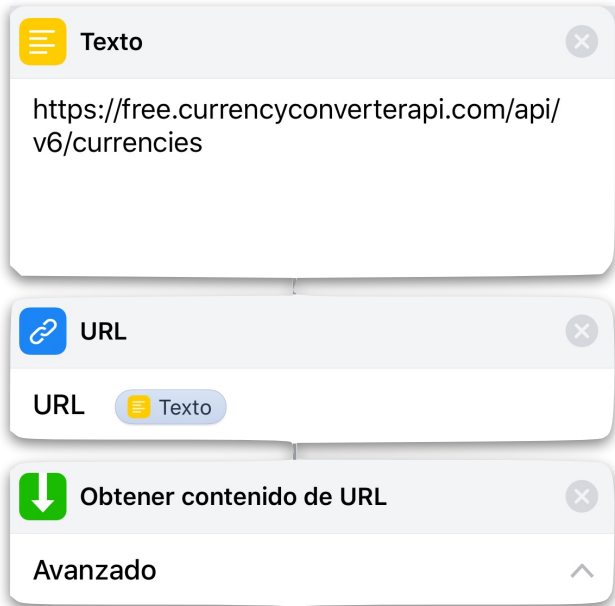


LOS VERBOS APARECEN EN PRIMER LUGAR EN MAYÚSCULA.

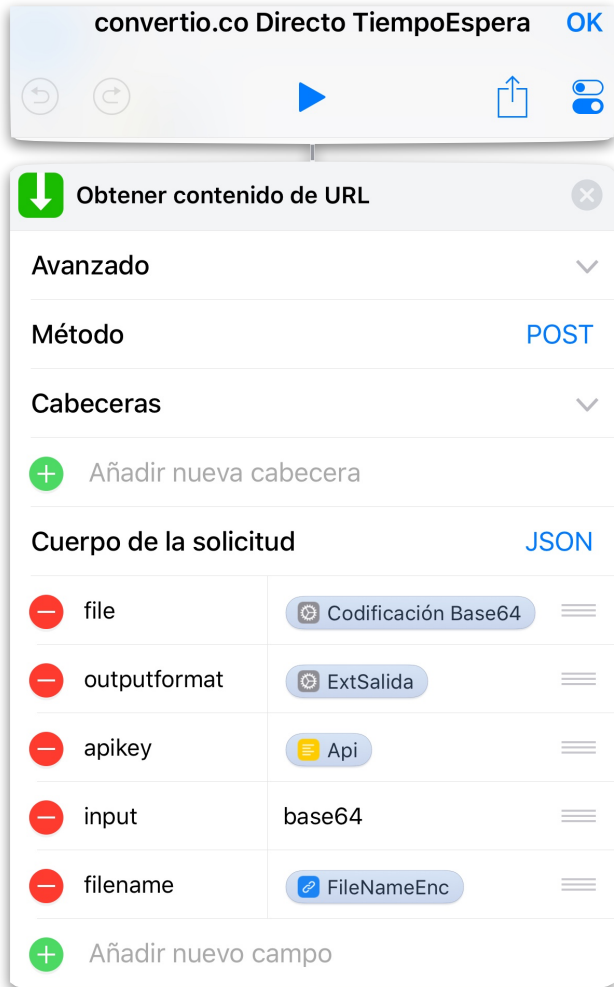
Después se puede configurar la petición, que dependerá de cada servicio al que estés llamando, lo que si se puede distinguir es entre **Cabeceras (HEADER)** y **Cuerpo de la solicitud(REQUEST BODY)**, opción que no aparece si el verbo es GET.

- El HEADER es un conjunto de metadatos sobre la petición (request) que se está enviando.
- El REQUEST BODY dependerá también de cada Web API, pero a grandes rasgos Shortcuts soporta 3 tipos de BODY REQUEST. El más común es JSON, pero admite también Formulario (BODY FORM) y Archivo (FILE). Por lo que respecta a JSON es un método clásico y el que usarás en casi todas las ocasiones, se basa en clave - valor, y es relativamente fácil de configurar. Archivo (FILE) lo que permite es, como puedes pensar, pasar un fichero para subirlo.

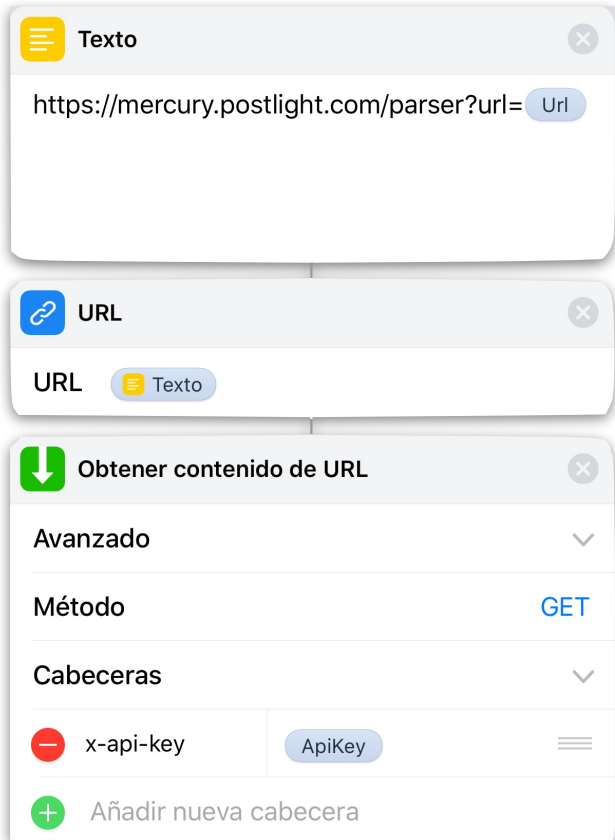
Antes de ver los ejemplos, se presentan a continuación algunas capturas de configuración de distintos servicios Web, con las diferentes opciones que se han abordado en este capítulo.



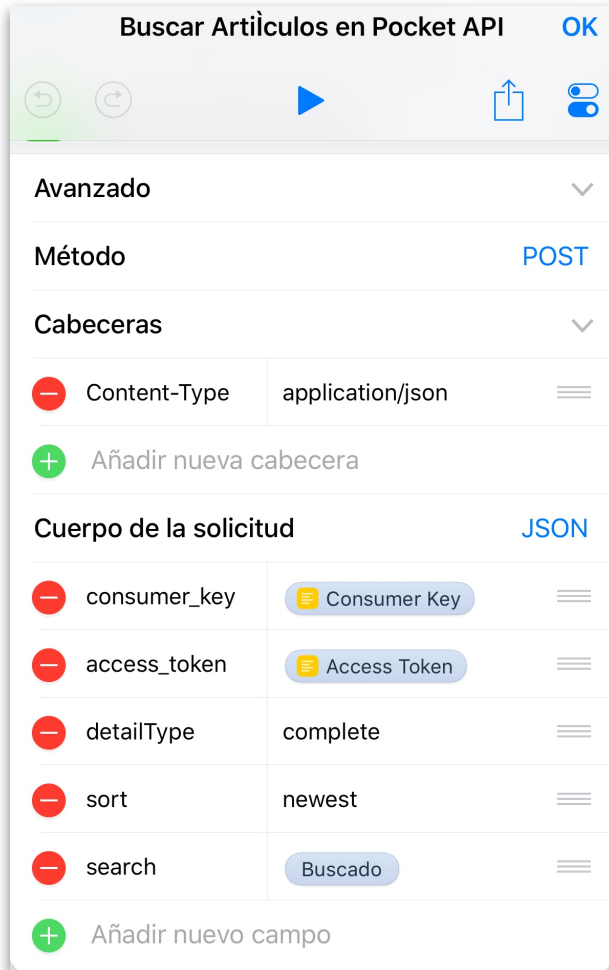
API SIMPLE



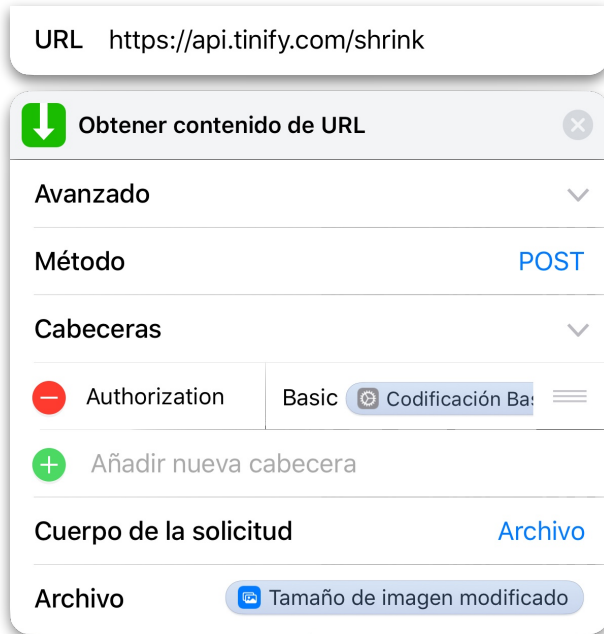
POST Y FICHERO EN BASE64



GET CON APIKEY Y PARÁMETRO (QUERY STRING)



POST EJEMPLO CON POCKET



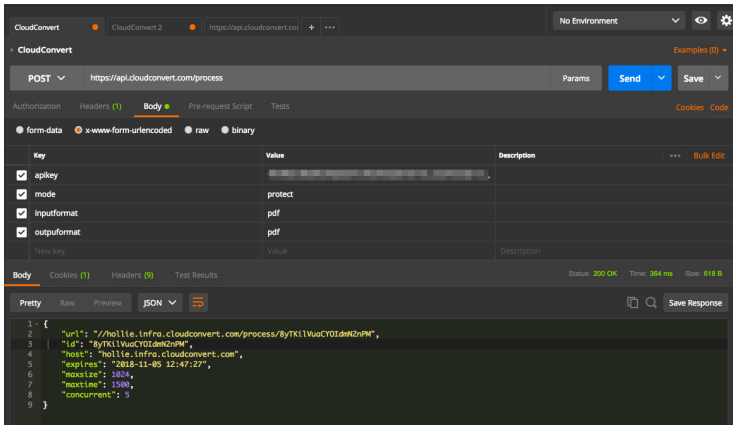
POST TINIFY CON CUERPO DE LA SOLICITUD DEL TIPO ARCHIVO

8.5.1. Herramientas - Postman

No es posible cerrar el capítulo sin nombrar una de las herramientas por excelencia en el uso de APIs: [Postman](#)

A grandes rasgos, Postman surgió originariamente como una extensión para el navegador Google Chrome, pero ya se pueden encontrar apps nativas para MAC, Windows y Linux.

Con esta herramienta, el testeo de Rest APIs es mucho más sencillo, así que es el primer lugar donde acudir si se resiste alguna llamada a una Web API en Shortcuts.

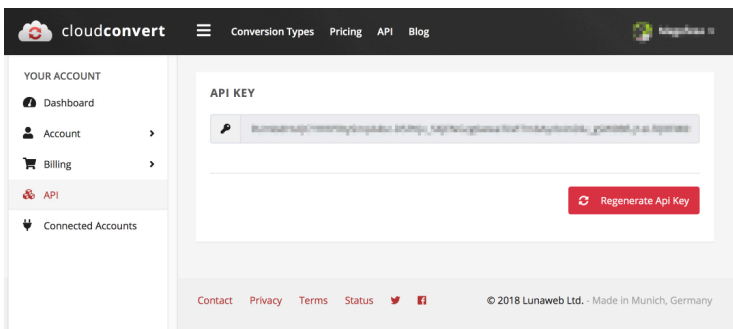


Para ejemplificar un poco más el uso de las API y usar esta fantástica herramienta, se van a presentar cuatro ejemplos. En ellos encontrarás la metodología para preparar shortcuts nivel pro.

8.5.1.1 Primer ejemplo

El primero se basa en utilizar la API de [CloudConvert](#) para poder **proteger un PDF mediante contraseña**. Los pasos a seguir para poder realizar este atajo son:

1. Registro en la web de CloudConvert de forma gratuita para conseguir una apiKey.



REGISTRO CLOUDCONVERT

2. Leer la documentación de la API*, identificar verbos y end points a aplicar. En este caso hay que hacer dos llamadas, una para crear el proceso, y la segunda que depende de una URL de la primera llamada y que hará la conversión, retornando un JSON donde hay una URL de descarga con el resultado.

Protect PDF

Live Test using API Console

Using the `protect` mode parameter, CloudConvert can encrypt PDF documents and add security restrictions.

Raw Request **cURL** PHP node.js Python

```
curl "https://srv01.cloudconvert.com/process/v4cw7zhf3" \
-F "mode=unlock" \
-F "input=download" \
-F "file=http://your_url/to.pdf" \
-F "converteroptions[encrypt_user_password]=password" \
-F "converteroptions[encrypt_owner_password]=password" \
-F "converteroptions[encrypt_allow_accessibility]=true" \
-F "converteroptions[encrypt_allow_extract]=false" \
-F "converteroptions[encrypt_allow_modify]=false" \
```

API PROTEGER PDF

3. Probar estas llamadas en Postman

CloudConvert Converter CloudConvert Protec X https://bepine.adr... No Environment

CloudConvert Proteger PDF

POST https://api.cloudconvert.com/process

Authorization Headers Body Pre-request Script Tests

form-data x-www-form-urlencoded **raw** binary

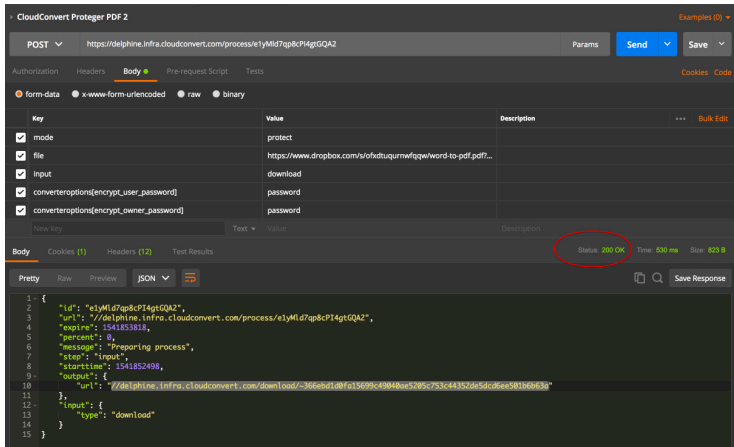
Key	Value	Description
apikey	[[[REDACTED]]]	
mode	protect	
inputformat	pdf	
outputformat	pdf	

Body Cookies Headers Test Results

Preview JSON

```
1: {
2:   "url": "://delphine.infra.cloudconvert.com/process/e1yM1d7sp8CP1qgt0A2",
3:   "id": "e1yM1d7sp8CP1qgt0A2",
4:   "host": "://delphine.infra.cloudconvert.com",
5:   "expires": "2018-11-11 00:21:24",
6:   "hostTime": 1814,
7:   "hostTime": 1330,
8:   "concurrent": 5
9: }
```


POSTMAN PROCESO




POSTMAN CONVERSIÓN

En este enlace podrás ver la aplicación en Postman, lógicamente no está la API por seguridad, por lo que deberías sustituirla por la tuya. [CloudConvert Postman](#)

4. Una vez comprobado, puedes aplicarlo a tus shortcuts.

 Texto ✕

Https://api.cloudconvert.com/process

 URL ✕

URL 

 Obtener contenido de URL ✕

Avanzado ▼

Método POST

Cabeceras ▼

 Añadir nueva cabecera

Cuerpo de la solicitud JSON

 apikey  ☰


 mode protect ☰

 inputformat pdf ☰

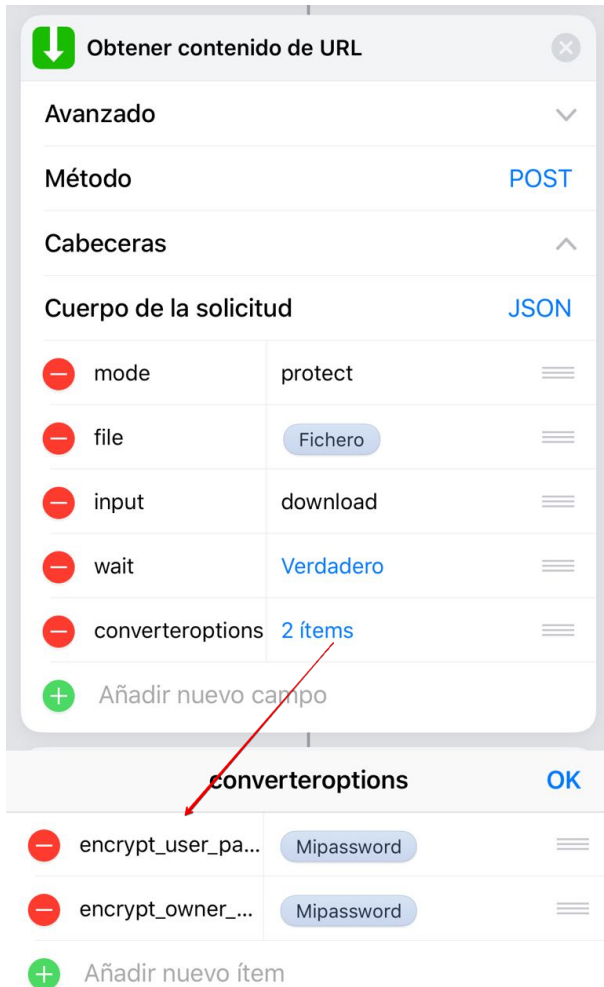
 outputformat pdf ☰

 Añadir nuevo campo

 Obtener diccionario de la entrada ✕

 Buscar

SHORTCUT - PROCESO



SHORTCUT - CONVERSIÓN



DESCÁRGALO AQUÍ

8.5.1.2 Segundo ejemplo

Siguiendo con CloudConvert, esta vez el objetivo es realizar un atajo que realice conversiones entre formatos.

1. Documentación de la API

Creating a new Process ID

In order to start a new conversion you need to create a new Process ID first. Each Process ID is for one-time use only and can only be used for converting one file.

Raw Request `cURL` `PHP` `node.js` `Python`

Example Request

```
POST https://api.cloudconvert.com/process
```

Authorization: Bearer API_KEY

```
{
  "inputformat": "flv",
  "outputformat": "mp4"
}
```

Parameter	Description
<code>inputformat</code>	* Current format of the file. (Required)
<code>outputformat</code>	* Output format, to which the file should be converted. (Required)
<code>mode</code>	The mode of the process. Defaults to <code>convert</code> . Other possible values are <code>info</code> , <code>combine</code> , <code>archive</code> and <code>extract</code> . (Optional)

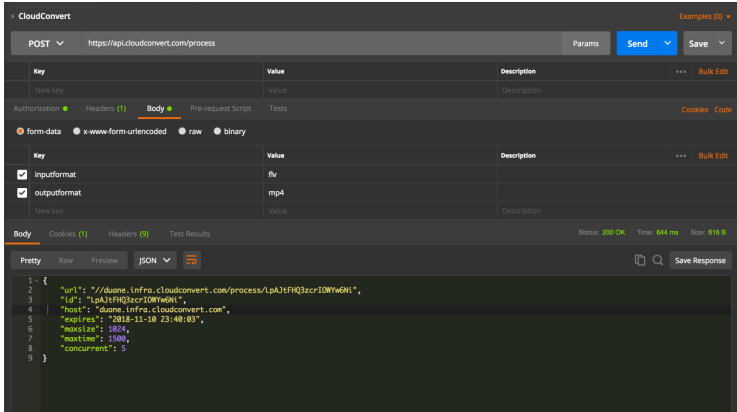
A response could look like this:

Example Response

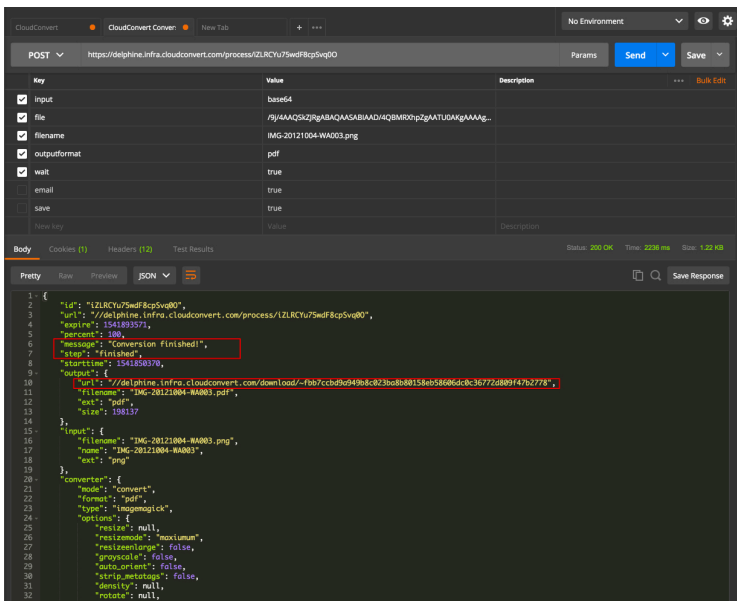
```
{
  "url": "//srv01.cloudconvert.com/process/v4cw72hf3",
  "id": "v4cw72hf3",
  "host": "srv01.cloudconvert.com",
  "expires": "2014-09-12 13:13:00",
  "maxtime": 18000,
  "minutes": 264
}
```

API CONVERSIÓN

2. Test en Postman



PROCESO




CONVERSIÓN

Enlace a la demo de la API [aquí](#)

3. Aplicar API en nuestro Atajo



PROCESO

 **Obtener contenido de URL** ✕

Avanzado ▾


Método POST

Cabeceras ▴

Cuerpo de la solicitud JSON

− input	<input type="text" value="Input"/>	☰
− file	<input type="text" value="File"/>	☰
− outputformat	<input type="text" value="ExtSalida"/>	☰
− email	<input type="text" value="true"/>	☰
− wait	<input type="text" value="true"/>	☰
− download	<input type="text" value="true"/>	☰
− filename	<input type="text" value="Nombre"/> <input type="text" value="Input File"/>	☰
− save	<input type="text" value="true"/>	☰

+ Añadir nuevo campo

 **Abrir en...** ✕

Mostrar menú "Abrir en"

CONVERTIR



[DESCÁRGALO AQUÍ](#)

8.5.1.3 tercer ejemplo

Un ejemplo más, en esta ocasión publicar en [Mastodon](#). Siguiendo el mismo proceso que anteriormente...

1. Documentación de la [API](#)

Media

Uploading a media attachment:

```
POST /api/v1/media
```

Form data:

Field	Description	Optional
file	Media to be uploaded (encoded using <code>multipart/form-data</code>)	no
description	A plain-text description of the media, for accessibility (max 420 chars)	yes
focus	Focal point: Two floating points, comma-delimited	yes

Returns an [Attachment](#) that can be used when creating a status.

Updating a media attachment:

```
PUT /api/v1/media/:id
```

Form data:

Field	Description	Optional
description	A plain-text description of the media, for accessibility (max 420 chars)	yes
focus	Focal point: Two floating points, comma-delimited	yes

Can only be done before the media is attached to a status. Returns an [Attachment](#) that can be used when creating a status.

Focal points

Server-side preview images are never cropped, to support a variety of apps and user interfaces. Therefore, the cropping must be done by those apps. To crop intelligently, focal points can be used to ensure a certain section of the image is always within the cropped view. [See this for how to let users select focal point coordinates.](#)

API DE MASTODON

2. Test en Postman

Mastodon 1

This test uses the "followed" API key and shows a basic OCI conversion for an image submitted via URL. Examples (0)

POST <https://mastodon.social/api/v1/media> Params Send Save

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary

Key	Value	Description
<input checked="" type="checkbox"/> file	<input type="text" value="Evo2_annonci_316345-3.jpg"/>	

Body Cookies (1) Headers (19) Test Results Status: 200 OK Time: 190 ms Size: 1.34 KB

```

1 {
2   "id": "7827633",
3   "type": "Image",
4   "url": "https://files.mastodon.social/media_attachments/files/007/827/613/original/7e44a280ef606c5e.jpg",
5   "preview_url": "https://files.mastodon.social/media_attachments/files/007/827/613/small/7e44a280ef606c5e.jpg",
6   "remote_url": null,
7   "text_url": "https://mastodon.social/media/7z9h9g1f1l7u4Lc584",
8   "meta": {
9     "original": {
10      "width": 300,
11      "height": 441,
12      "size": "300x441",
13      "aspect": 0.688221888453374
14    },
15    "small": {
16      "width": 300,
17      "height": 441,
18      "size": "300x441",
19      "aspect": 0.688221888453374
20    }
21  },
22   "description": null
23 }

```

AUTENTICACIÓN

Mastodon 2

POST <https://mastodon.social/api/v1/statuses> Params Send Save

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary

Key	Value	Description
<input checked="" type="checkbox"/> status	<input type="text" value="L2hYvcrqj0RZMfC6MDHJWwCCDMSDMEFwKvKw"/>	
<input checked="" type="checkbox"/> media_ids[]	<input type="text" value="7827633"/>	
<input checked="" type="checkbox"/> spoiler_text	<input type="text" value="NSFW - inapropiado"/>	

Body Cookies (1) Headers (19) Test Results Status: 200 OK Time: 118 ms Size: 2.8 KB

```

6   "visibility": "public",
7   "language": null,
8   "url": "https://mastodon.social/users/botari0a2x100/statuses/18184774894947451",
9   "content": "evolucionariopartidariosaoccuradosestefwckvkw",
10  "url": "https://mastodon.social/@botari0a2x100/18184774894947451",
11  "reply_count": 0,
12  "reblog_count": 0,
13  "favorited_count": 0,
14  "favourite": false,
15  "reblogged": false,
16  "retweeted": false,
17  "pinned": false,
18  "reply": null,
19  "application": {
20    "name": "Shortcuts",
21    "website": ""
22  },
23  "account": {
24    "id": "7823329",
25    "username": "botari0a2x100",
26    "acct": "botari0a2x100",
27    "display_name": "",
28    "locked": false,
29    "bot": false,
30    "created_at": "2018-11-05T00:51:32.366Z",
31    "note": "evolucionariop",
32    "url": "https://mastodon.social/@botari0a2x100",
33    "avatar": "https://mastodon.social/avatars/original/missing.png",
34    "avatar_static": "https://mastodon.social/avatars/original/missing.png",
35    "header": "https://mastodon.social/headers/original/missing.png",
36    "header_static": "https://mastodon.social/headers/original/missing.png",
37    "following_count": 0,
38    "followers_count": 0,
39    "statuses_count": 1,
40    "media_attachments": [
41      {
42        "id": "7827633",
43        "type": "Image",
44        "url": "https://files.mastodon.social/media_attachments/files/007/827/613/original/7e44a280ef606c5e.jpg",
45        "preview_url": "https://files.mastodon.social/media_attachments/files/007/827/613/small/7e44a280ef606c5e.jpg",
46        "remote_url": null,
47        "text_url": "https://mastodon.social/media/7z9h9g1f1l7u4Lc584",
48        "meta": {
49          "original": {

```

PUBLICACIÓN

Enlace a la demo de la API [aquí](#)

3. Aplicar API en el atajo

The screenshot displays two overlapping windows from an API client. The top window, titled 'URL', shows the endpoint: `https://Diccionario instancia /api/v1/m`. The bottom window, titled 'Obtener contenido de URL', is expanded to show the request configuration. It includes a dropdown for 'Avanzado', a 'Método' set to 'POST', and a 'Cabeceras' section with two headers: 'idempotency_...' with value 'idempotency_key' and 'Authorization' with value 'Bearer Diccionario'. Below the headers is an option to 'Añadir nueva cabecera'. The 'Cuerpo de la solicitud' section is set to 'Formulario' and contains a 'file' field with a 'Fotos' selection. An option to 'Añadir nuevo campo' is also visible.

URL `https://Diccionario instancia /api/v1/m`

Obtener contenido de URL

Avanzado

Método **POST**

Cabeceras

idempotency_... idempotency_key

Authorization Bearer Diccionario

+ Añadir nueva cabecera


Cuerpo de la solicitud **Formulario**

file Fotos

+ Añadir nuevo campo

URL ✕



URL <https://Diccionario instancia /api/v1/st>


 **Obtener contenido de URL** ✕

Avanzado ▾



Método POST

Cabeceras ▾

	idempotency_...	idempotency_key	
	Authorization	Bearer  Diccionario	

 Añadir nueva cabecera

Cuerpo de la solicitud Formulario

	spoiler_text	NSFW - Inapropiado	
	sensitive	Texto	

2



[DESCÁRGALO AQUÍ](#)

8.5.1.4 Cuarto ejemplo

Para finalizar esta serie de ejemplos, un último shortcut que realiza OCR a un documento dado.

1. Registro gratuito para acceder a la API en [OCR.Space](#)

Getting Started	Getting Started
Get your free API key	API Basics
Free OCR API Endpoint	The free OCR API provides a simple way of parsing images and multi-page PDF documents (PDF OCR) and getting the extracted text results returned in a JSON format. The API can be used from any internet-connected device (desktop, mobile, iPhone, Android, Windows phone, refrigerator...). For even faster response times and guaranteed uptime PRO plans are available. In addition, we offer a locally (on-premise) installable version with our OCR.space Local Enterprise plan.
Ordering a PRO Plan	Get Your Free OCR API Key
On-Premise OCR	The OCR API has three tiers/levels. If you want to use the free tier register here for your free OCR API key . It has a rate limit of 500 requests within one day per IP address to prevent accidental spamming.
GET Parameters	
POST Parameters	
Response	
Searchable PDF	
Code Examples	
C#	
C++/QT	
cURL	
Java (Android app)	
Javascript/Query	
PHP	
Python	
Ruby	
Swift/Objective-C (iPhone)	

OCR API Tier	Free	PRO	PRO PDF
Pricing	Free	US\$24.95/month	US\$49.95/month
Requests/month	25,000	250,000	250,000
Additional conversions	n/a	US\$10/100,000	US\$20/100,000
API Key	Register for free API key	Buy PRO API Key	Buy PRO PDF API Key
File Size Limit	1 MB	5 MB	100 MB+
PDF OCR Page Limit	3	(images only)	999+
Make Searchable PDF	Yes (with watermark)	Yes, from images	Yes
Commercial use allowed	Yes	Yes	Yes
Speed	Fast	Faster (more servers, lower load)	
Rate Limit*	500 calls/DAY	600 calls/1min	600 calls/1min
Service-level agreement (SLA)	n/a	100% uptime or money back (dedicated, redundant servers in USA/EU/Asia)	

REGISTRO

2. Leer Documentación API

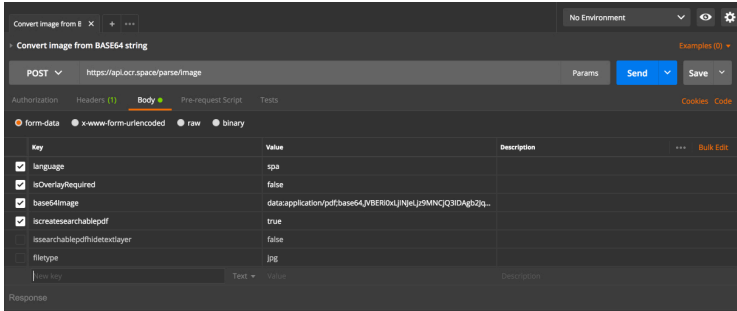
Post Parameters

The table below lists all possible API parameters. As additional documentation, we [published](#) a sample api call collection that you can load into [Postman](#). And last but not least: Our [free online ocr](#) form on the front page is nothing else than a POST call to the free OCR API endpoint and can be used for testing as well.

Key	Value	Description
<code>apikey</code>	API Key (send in the header)	Get your free API key
<code>url</code> or <code>file</code> or <code>base64Image</code>	<code>url</code> : URL of remote image file (Make sure it has the right content type) <code>file</code> : Multipart encoded image file with filename <code>base64Image</code> : Image as Base64 encoded string	You can use three methods to upload the input image or PDF. We recommend the URL method for file sizes > 10 MB for faster upload speeds.
<code>language</code>	[Optional] Arabic= <code>ara</code> Bulgarian= <code>bul</code> Chinese(Simplified)= <code>chs</code> Chinese(Traditional)= <code>cht</code> Croatian = <code>hrv</code> Czech = <code>cze</code> Danish = <code>dan</code> Dutch = <code>dut</code> English = <code>eng</code> Finnish = <code>fin</code> French = <code>fre</code> German = <code>ger</code> Greek = <code>gre</code> Hungarian = <code>hun</code> Korean = <code>kor</code> Italian = <code>ita</code> Japanese = <code>jpn</code> Norwegian = <code>nor</code>	Language used for OCR. If no language is specified, English <code>eng</code> is taken as default. IMPORTANT: The language code has always 3-letters (not 2). So it is "eng" and not "en".

DOCUMENTACIÓN

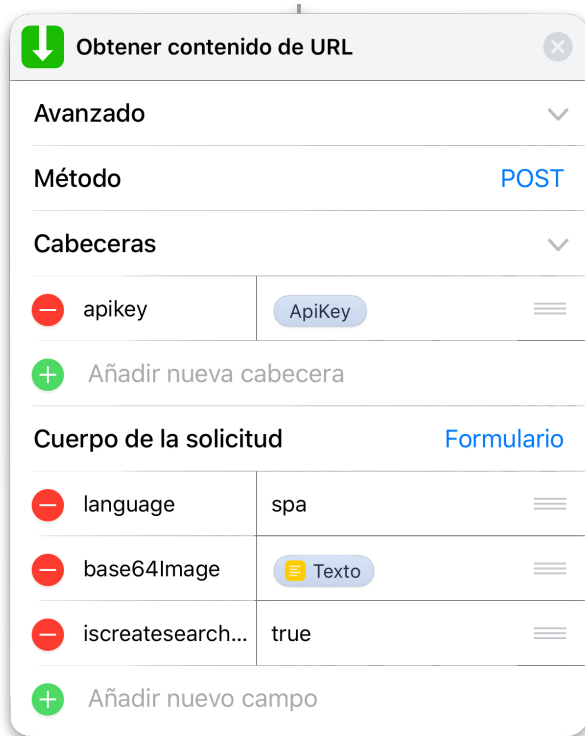
3. Test en Postman



POSTMAN OCR

Enlace [aquí](#)

4- Aplicar en el shortcut



SHORTCUT



[DESCÁRGALO AQUÍ](#)

Hasta aquí este capítulo dedicado a las Web APIs y su uso en Shortcuts. Como has podido comprobar, “*Obtener Contenido URL*” va mucho más allá de lo que parece. Lo importante es que entiendas que todo lo fácil que “parece” al arrastrar ese bloque en Shortcuts, esconde detrás conceptos que provocarán a menudo dudas y resultados no esperados (errores HTTP 400). Sólo la paciencia y la experiencia te ayudarán a configurar servicios.

9. Como ejecutar los shortcuts

Aunque en el punto 2.3.2 de este libro se habló de cómo se puede lanzar la ejecución de los shortcuts, ahora se va a profundizar en la ejecución e integración con ciertos programas que tienen una relación más estrecha son Shortcuts, como son recordatorios, Launch Center Pro, Launcher, DUE, o IFTTT.

9.1. Recordatorios

La app recordatorios no necesita presentación, viene preinstalada en el sistema operativo desde iOS 5, que apareció en 2011.

Apple, desde iOS 11, permite borrar las aplicaciones del sistema, como recordatorios. Si en tu caso has borrado **recordatorios** recuerda que puedes volver a instalarla buscándola en la App Store.

A priori, la app recordatorios no permite adjuntar shortcuts, pero si se lo pides a Siri de una forma específica si que puedes hacerlo.

Para que todo funcione perfectamente deberás tener activadas las notificaciones para recordatorios como se ve en la siguiente imagen:

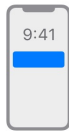
[← Atrás](#)

Recordatorios

Permitir notificaciones



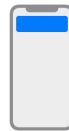
AVISOS



Pantalla bloqueada



Centro de
notificaciones



Tiras



Estilo de tira

Permanente >

Sonidos

Acorde >

Globos



OPCIONES

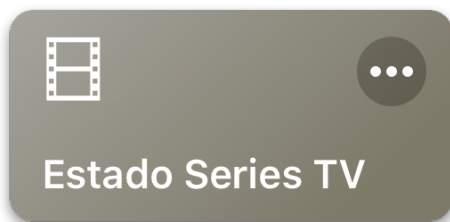
Mostrar previsualizaciones Si está desbloque... >

Agrupar notificaciones

Automático >

AJUSTES DE NOTIFICACIONES PARA LA APP RECORDATORIOS

Ahora abre la aplicación Shortcuts (o Atajos, en español) y abre el shortcut que quieras que se te recuerde pulsando en el círculo con los tres puntos que contiene su icono arriba a la derecha:



EL BOTÓN PARA ABRIR EL SHORTCUT SIN EJECUTARLO SON LOS TRES PUNTOS QUE HAY DENTRO DEL CÍRCULO EN LA PARTE SUPERIOR DERECHA DEL ICONO DEL SHORTCUT

Ahora invocamos a Siri de la siguiente manera según lo que quieras conseguir:


- “Oye Siri, recuérdame este atajo dentro de 5 minutos”
- “Oye Siri, recuérdame este atajo todos los días a las 2 de la tarde”
- “Oye Siri, recuérdame este atajo mañana a las 8 de la tarde”
- “Oye Siri, recuérdame este atajo cuando llegue al trabajo”
- “Oye Siri, recuérdame este atajo cuando salga de esta ubicación”



Estos son solo un ejemplo, puedes idear la combinación que más se ajuste a tus necesidades, eso si, el comienzo, como puedes ver es siempre el mismo “Oye Siri, recuérdame este atajo ...”.

Oye Siri recuérdame esta Tajo
dentro de un minuto

Pulsa para editar >

De acuerdo. He añadido "" a tus recordatorios. Vence dentro de 1 minuto:

 RECORDATORIOS


 Estado Series TV
Hoy 13:41 

Eliminar



ESTA SERÁ LA RESPUESTA DE SIRI SI TODO ES CORRECTO.

Cuando se cumpla la condición que le has indicado a Siri aparecerá una notificación en tu teléfono o Apple Watch como la de la siguiente imagen. Con solo pulsar la notificación se abrirá recordatorios y a la derecha se mostrará el icono de Shortcuts, al hacer clic sobre él se abrirá directamente el shortcut que tenias abierto en el momento de invocar a Siri.

 RECORDATORIOS ahora

Estado Series TV



 RECORDATORIOS ⋮

Estado Series TV

Abrir Atajos


Completado

Recordar dentro de 1 hora

Recordar mañana

Recordatorios 1

iPhone Editar

Estado Series TV 

Hoy 13:08

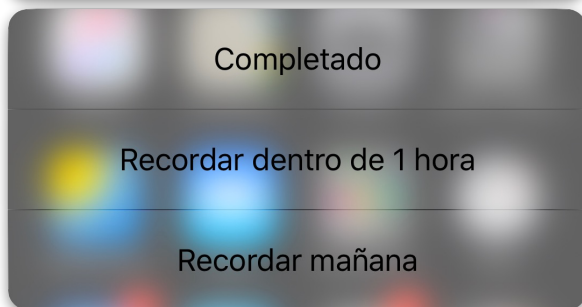
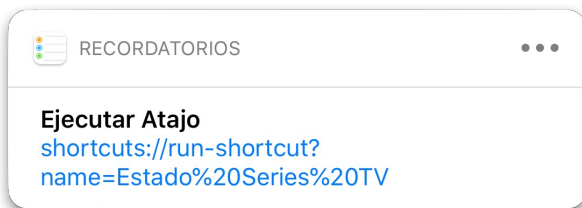
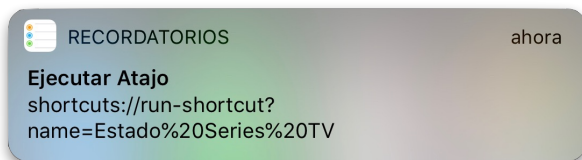
+

SECUENCIA DE LA NOTIFICACIÓN. 1. LA NOTIFICACIÓN APARECE A LA HORA O LUGAR PROGRAMADO. 2. EL MENÚ QUE SE ABRE AL TIRAR HACIA ABAJO DE LA NOTIFICACIÓN. 3. SE ABRE LA APP DE RECORDATORIOS Y PUEDE VERSE EL ICONO DE SHORTCUTS A LA DERECHA.

Otra manera de añadir un shortcut concreto en un recordatorio es usar los url-schemes de Shortcuts con la siguiente url:

```
shortcuts://run-shortcut?  
name=NombreDelShortcut
```

De esta manera al usar 3D Touch, es decir, presionando fuerte sobre la notificación podrás ejecutar directamente el shortcut que coincida con el nombre que has especificado. En este caso si que se ejecuta directamente el shortcut.



SECUENCIA DE LA NOTIFICACIÓN. 1. LA NOTIFICACIÓN APARECE A LA HORA O LUGAR PROGRAMADO. 2. EL MENÚ QUE SE ABRE AL TIRAR HACIA ABAJO DE LA NOTIFICACIÓN O PULSAR FUERTE SOBRE LA NOTIFICACIÓN. 3. AL PINCHAR SOBRE EL ENLACE SE EJECUTA EL SHORTCUT DIRECTAMENTE

Recuerda que al tratarse de una URL los espacios que tenga el nombre del shortcut deberás sustituirlos por %20. En este caso concreto queda así

```
shortcuts://run-shortcut?  
name=Estado%20Series%20TV
```

La gran ventaja de este sistema, que es un poco mas engorroso, es que la ejecución del shortcut es directa, por tanto, se puede concluir que si necesitas un recordatorio puntual es más rápido usar Siri y si es un recordatorio repetitivo merece la pena tardar un poco más a cambio de que la ejecución sea directa.

9.2. IFTTT

[IFTTT](#) es un servicio gratuito en la nube para hacer que dispositivos y servicios de distintas marcas puedan entenderse y hacer cosas juntos.

IFTTT es el acrónimo de *If This Then That* que en español es *Si Esto Entonces eso*. Su nombre, por tanto, habla por si mismo del funcionamiento de esta web: eliges dos servicios o dispositivos, si en uno ocurre algo, en el otro pasa lo que hayas previsto.

Un ejemplo sencillo: conectas tu cuenta de Gmail y una bombilla inteligente [Philips Hue](#), la receta, que es como llaman ellos a sus acciones, es “Si gmail recibe un correo entonces Philips Hue se enciende”.

También admite servicios y redes sociales como por ejemplo: “Si nueva fotografía en Instagram guardala en dropbox”.

Por todo esto, se ha convertido en un imprescindible silencioso para muchos.

En el caso del antiguo Workflow, había una integración directa entre ambas aplicaciones, pero con la salida de Shortcuts desapareció. A pesar de todo, esto no es un problema, ya que desde hace tiempo IFTTT cuenta con un servicio genérico desde el que dar servicio a dispositivos que tengan API o URL-SCHEME llamado WebHooks.

¿Qué es un Webhook? Al contrario que las APIs en las que el A hace una petición de información a B, las WebHooks permiten a B enviarle información a A sin que esta la pida. En este caso A estará oyendo por si B le habla.

Las aplicaciones, cada día más, necesitan comunicarse entre ellas, para ello existen las REST APIs, llamadas a APIs etc... Lo importante es saber que un webhook se puede definir como la manera de que una app, pueda “suministrar” información a una o más apps en tiempo real. El comportamiento de las API siempre ha sido bajo petición (Polling), mientras que en este caso es la aplicación quien “notifica” al resto que se ha producido una acción determinada, por eso se las llama “Reverse APIs”(APIs Invertidas).

Al “mensaje” que envía la primera app a través de una URL única de le llama “payload”, y se puede entender como una dirección o número de teléfono.

Imagina un ejemplo sencillo, dos personas están en la misma habitación, se llaman Paloma y Maite. Paloma tendrá algo que decirle a Maite en algún momento, pero Maite no sabe cuando. Hay dos opciones:

- Maite pregunta cada 5 minutos a Paloma si tiene algo que decirle.
- Maite está atenta y cuando Paloma tenga algo que decir lo dirá en voz alta, por lo que Maite recibirá la

información. Es más, si Inma está también en la habitación, también recibe la información.

El primer caso es el funcionamiento de una API, el segundo el de un WebHook.

Por ahora la comunicación no va en ambos sentidos, por ahora cuando ejecutes un shortcut este hará de lanzador para que ocurra otra acción dentro de IFTTT, en este caso el shortcut estaría en la parte **This**.

Aprenderás a configurarlos para que funcionen perfectamente junto con tus shortcuts de dos maneras:

- Notificaciones enriquecidas
- WebHooks

9.2.1 Lanzar Shortcuts a partir de una notificación de un evento en IFTTT

Hasta aquí se ha visto como usar IFTTT desde un shortcut. Ahora vas a ver como a partir de un evento en IFTTT, por ejemplo llegas a un determinado lugar ([geovalla](#)) y quieres ejecutar un shortcut.

Para ello, se usará un receta del tipo:

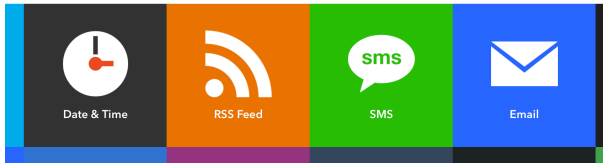
SI {evento} ENTONCES {NOTIFICATION}

Selecciona primer servicio (trigger), en este caso usa "Date & Time"

Choose a service

Step 1 of 6

Q Search services



BUSCA "DATE & TIME"

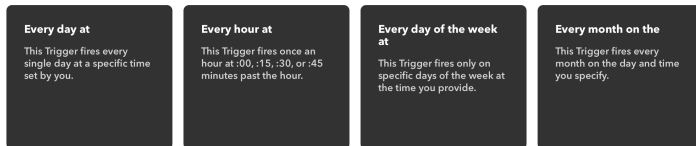
Luego selecciona "Every day at" para lanzar esta acción todos los días a la misma hora.

← Back



Choose trigger

Step 2 of 6



Usa las 12 am o la hora que mas te interese para ver que funciona en este momento, por ejemplo dentro de 5 minutos.




Complete trigger fields


Step 2 of 6

Every day at

This Trigger fires every single day at a specific time set by you.

Time

12 AM 

00 Minutes 

Create trigger

CONFIGURA LA HORA QUE MAS TE CONVENGA.

Selecciona el “THAT” y busca el servicio “NOTIFICATIONS”

Usa la opción “Send a rich notification from the IFTTT app”, por supuesto esta opción requiere tener la aplicación de IFTTT instalada en el dispositivo.

◀ Back



Choose action

Step 4 of 6

Send a notification from the IFTTT app

This action will send a notification to your devices from the IFTTT app.

Send a rich notification from the IFTTT app

This action will send a rich notification to your devices from the IFTTT app. Rich notifications may include a title, image, and link that opens in a browser or installed app.

ASEGURATE DE HABER SELECCIONADO EL SERVICIO “RICH NOTIFICATIONS”



Complete action fields

Step 5 of 6

Send a rich notification from the IFTTT app

This action will send a rich notification to your devices from the IFTTT app. Rich notifications may include a title, image, and link that opens in a browser or installed app.

Title

Optional, bold text above the message [Add ingredient](#)

Message

[Add ingredient](#)

Link URL

Optional, link may open in a browser or installed app [Add ingredient](#)

Image URL

Optional [Add ingredient](#)

[Create action](#)

CONFIGURAR TÍTULO, CUERPO DEL MENSAJE.

En el campo “Link URL” definir la URL del modo:

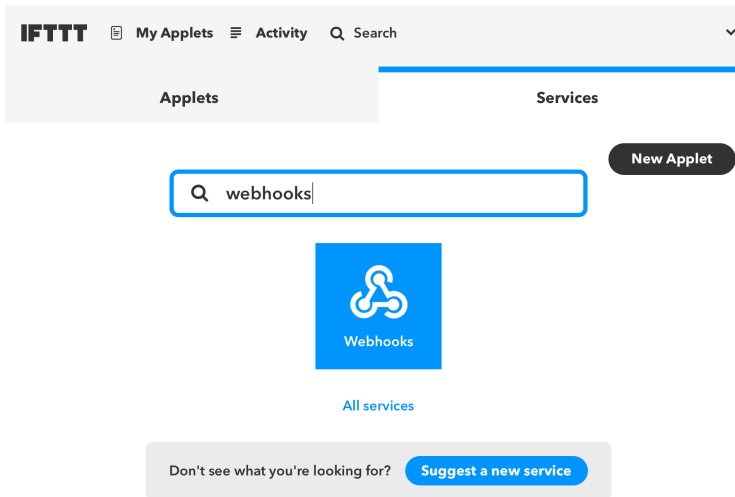

```
`shortcuts://run-shortcut?
name={Nombre del Shortcuts}`
```

Si se desea, se puede hasta configurar un enlace a una imagen para hacerlo más visual.

Entonces, cuando actúe el disparador (trigger), IFTTT te pedirá permiso para abrir Shortcuts y ejecutará las acciones que hayan en él.

9.2.2 Peticiones a un servicio a través de WebHooks

En primer lugar dirígete a la web de IFTTT y date de alta si no lo habías hecho ya. A continuación en la pantalla principal pulsa sobre “Services” y en la barra de búsqueda escribe “webhooks” para que aparezca el servicio.



PANTALLA DE BÚSQUEDA DE SERVICIOS EN IFTTT

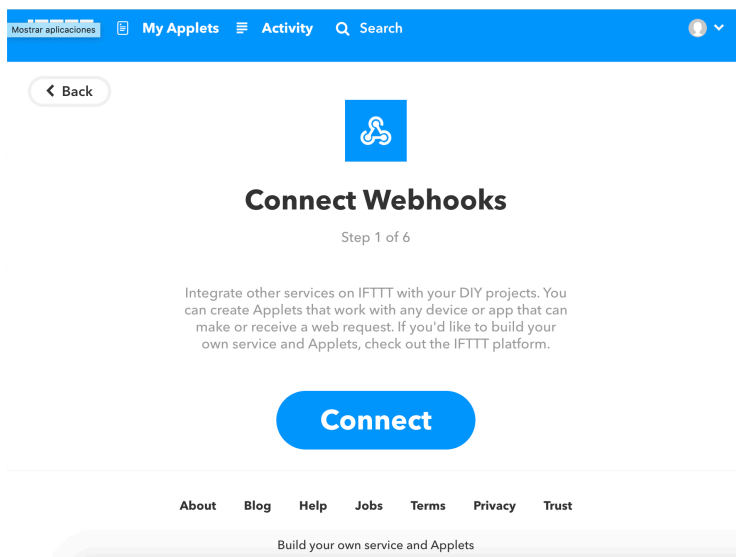
9.2.2.1 Caso en el que el shortcut es el lanzador

9.2.2.1.1 Configuración en IFTTT

En este caso, como se explicaba antes, el shortcut será la parte **This** de la receta de IFTTT, por lo que lanzará una respuesta que se haya configurado.

Ve a la página para crear una nueva receta en <https://ifttt.com/create>.

En primer lugar, tras buscar el servicio aparecerá un asistente que guiará la configuración. El paso 1 de 6 indica que con webhooks se puede integrar IFTTT con cualquier proyecto personal.



PASO 1 DE 6 DE LA CONFIGURACIÓN

Al pulsar sobre “Connect” se abre una ventana dónde deberás seleccionar el recuadro “Receive a web request”

[← Back](#)

Choose trigger

Step 2 of 6

Receive a web request

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

PASO 2 DE 6 DE LA CONFIGURACIÓN

En “Event Name” deberás introducir el nombre que quieras para identificar lo que va a ocurrir con este lanzador. En este ejemplo se va a lanzar una acción que mande un correo electrónico al ejecutar un shortcut. Por tanto lo puedes nombrar como shortcut-mail.

[← Back](#)

Complete trigger fields

Step 2 of 6

Event Name

The name of the event, like "button_pressed" or "front_door_opened"

Create trigger

PASO 3 DE 6 DE LA CONFIGURACIÓN

Se pulsa en "Create trigger" y a continuación queda todo preparado para configurar la segunda parte de la acción, el "that" que enviará el correo.

[← Back](#)[About](#) [Blog](#) [Help](#) [Jobs](#) [Terms](#) [Privacy](#) [Trust](#)

Build your own service and Applets

SE HA TERMINADO DE CONFIGURAR EL “THIS”

Busca y selecciona el servicio “Email” y dentro usa la acción “Send me an email”.

← Back



Choose action

Step 4 of 6

Send me an email

This Action will send you an HTML based email. Images and links are supported.

PASO 4 DE 6 DE LA CONFIGURACIÓN

Por defecto el asunto del mensaje es este: > The event named “{{EventName}}” occurred on the Maker Webhooks service

Igual que en la realidad las recetas necesitan ingredientes, en IFTTT las recetas tienen disponibles ingredientes, es decir, marcadores de datos que se usan para que podamos enviar y recibir información a través de la comunicación en IFTTT entre el lanzador y la acción.

En el caso de los **Webhooks** los ingredientes son: * **EventName**: **recoge el nombre que se le ha dado al lanzador.** * **Value1**, **Value2**** y **Value3**: son valores genéricos que puedes usar para mandar los datos que necesites. * **OccuredAt**: recupera automáticamente la fecha de la ejecución y la añade al mensaje.

Como se ha dicho, {{EventName}} es, por tanto, un marcador de posición que se sustituye en el momento del envío por el

nombre que se le ha dado en el momento de crear el disparador, en este caso shortcut-mail.

Se puede sustituir por el asunto que quieras, para este ejemplo usa: El shortcut ha funcionado correctamente

El cuerpo de mensaje (Body) se configurará con este texto:

```
{{OccurredAt}}<br><br>
El atajo ha usado el lanzador {{EventName}}
para comunicarse con IFTTT y ha funcionado
correctamente.
Además hemos aprovechado para enviar estos
datos: {{Value1}}, {{Value2}}, {{Value3}}.
```

Step 5 of 6

Subject

El shortcut ha funcionado correctamente

Add ingredient

Body

```
{{OccurredAt}}<br><br>
El atajo ha usado el lanzador
{{EventName}} para comunicarse
con IFTTT y ha funcionado
correctamente.
Además hemos aprovechado para
enviar estos datos: {{Value1}},
{{Value2}}, {{Value3}}.
```

Add ingredient

Create action

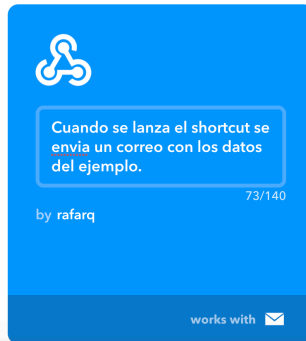
PASO 5 DE 6 DE LA CONFIGURACIÓN

Al pulsar sobre Create action aparece el último paso donde podrás escribir un texto explicativo de qué hace esa receta, de manera que en el futuro recuerdes que has hecho.

← Back

Review and finish

Step 6 of 6



PASO 6 DE 6 DE LA CONFIGURACIÓN

Pulsando en “Finish” se termina la configuración web de la receta.



Cuando se lanza el shortcut se envia un correo con los datos del ejemplo.

by  rafarq

[Learn more](#)

On



- Created on Dec 12 2018
- Never run

This Applet usually runs within a few seconds

[Check now](#)

works with 

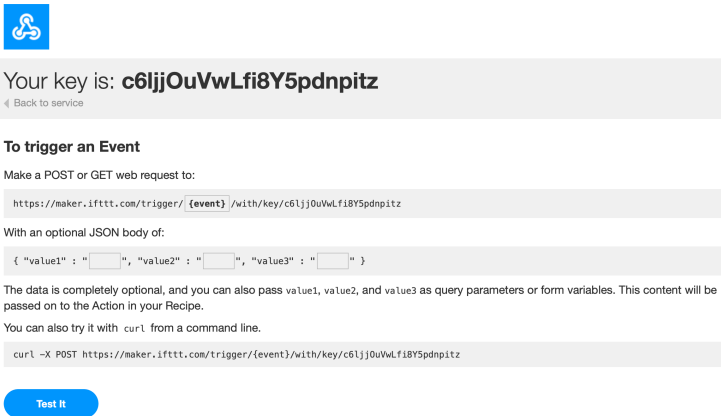
ESTE ES EL RESULTADO DE LA RECETA


9.2.2.1.2 Funcionamiento en Shortcuts

Para mostrar el funcionamiento de la API de IFTTT Webhooks se creará un shortcut sencillo que funcionará entorno a la petición POST de la API.

Para obtener los datos de acceso a la API ve la raíz de webhooks https://ifttt.com/maker_webhooks

Verás una pantalla como esta:





Your key is: **c6ljjOuVwLfi8Y5pdnpitz**
[← Back to service](#)

To trigger an Event

Make a POST or GET web request to:

```
https://maker.ifttt.com/trigger/{event}/with/key/c6ljjOuVwLfi8Y5pdnpitz
```

With an optional JSON body of:

```
{ "value1" : " ", "value2" : " ", "value3" : " " }
```

The data is completely optional, and you can also pass value1, value2, and value3 as query parameters or form variables. This content will be passed on to the Action in your Recipe.

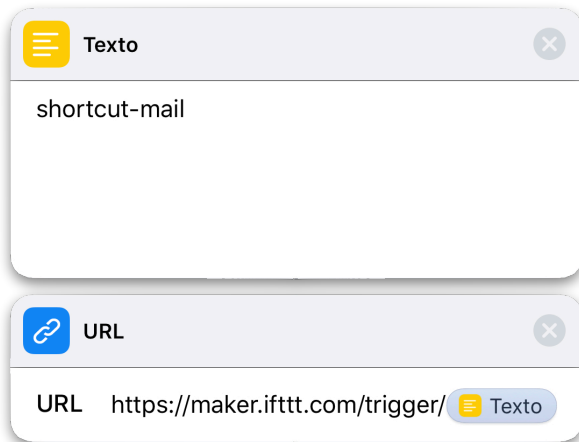
You can also try it with `curl` from a command line.

```
curl -X POST https://maker.ifttt.com/trigger/{event}/with/key/c6ljjOuVwLfi8Y5pdnpitz
```

[Test it](#)

ESTE ES EL RESULTADO DE LA RECETA

La URL principal ya está preparada con todo lo que necesitas para insertarla directamente en una acción de URL de tu shortcut. Solo queda sustituir el campo {event} por el nombre del evento que hemos configurado, recuerda: shortcut-mail.



CONFIGURACIÓN DE LA PETICIÓN A LA API

Una vez tienes la URL bien configurada solo queda configurar una acción del tipo “Obtener contenidos de URL” pulsando sobre avanzado para mostrar la configuración completa de la siguiente manera: * Método: POST * Cuerpo de la solicitud: JSON

Se añaden tres valores de tipo texto con el nombre value1, value2 y value3. A estos tres campos le puedes dar el valor que quieras o necesites para el funcionamiento correcto del shortcut, en este caso, para ser prácticos se han dado estos valores:

- value1: Texto del valor 1
- value2: Texto del valor 2
- value3: Texto del valor 3

Por tanto la acción “Obtener contenidos de URL” debe quedar así:



CONFIGURACIÓN DE LA URL EN SHORTCUTS

Una vez ejecutado debes recibir un correo de este tipo:

December 12, 2018 at 06:51PM

El atajo ha usado el lanzador shortcut-mail para comunicarse con IFTTT y ha funcionado correctamente. Además hemos aprovechado para enviar estos datos: Texto del valor 1, Texto del valor 2, Texto del valor 3.



Cuando se lanza el shortcut se envía un correo con los datos del ejemplo.



[Unsubscribe from these notifications](#) or sign in to manage your [Email Applets](#).

IFTTT

EMAIL RECIBIDO CON LOS DATOS QUE SE HAN ENVIADO DESDE EL SHORTCUT

9.3. Launch Center Pro / Launcher

Estas dos aplicaciones tienen parecidos más allá del propio nombre. [Launch Center Pro](#) es de Contrast Apps LLC, y [Launcher](#) de Cromulent Labs, Inc.

Si se habla de ellas en este libro es, precisamente, porque suplen, de alguna manera, carencias que Apple aún no ha resuelto en Shortcuts y, además, conocerlas y usarlas puede que te ayude en otros aspectos que no son objeto de Shortcuts y por tanto de este libro, pero que, sin duda, mejorarán la automatización que puede hacerse en iOS.

Las similitudes de ambas son las siguientes:

- permiten crear una serie de accesos directos a distintas acciones cotidianas en los dispositivos con iOS.
- permiten configurar un widget con esos accesos directos para tenerlos a mano.
- se pueden usar URL-SCHEMES para interactuar con otras apps.
- disponen de ayuda para la creación de las acciones.
- permiten configurar disparadores (triggers en inglés) para lanzar dichos accesos:

- disparadores de ubicación
 - disparadores de hora
- permiten ordenar los accesos en la pantalla y en carpetas.
- disponen de un amplio catálogo de aplicaciones con las que pueden integrarse.



DISPARADORES DISPONIBLES PARA LAS ACCIONES DE LAUNCHER

En cuanto a las diferencias:

Launcher, permite añadir un Siri Shortcut para cada acción. Además dispone de: * *launchers mágicos*, cuyos iconos son dinámicos y pueden mostrar información actualizada al momento. * *meta launchers*, que son capaces de crear launchers normales sobre la marcha.

Launcher usa URL-SCHEMES de forma transparente para el usuario, por lo que la interfaz mostrará las opciones para que las rellenes pero no tendrás la sensación, desagradable para algunos, de estar trabajando con código.

Launch Center pro, dispone de un lenguaje propio a la hora de crear acciones muy fácil de aprender una vez creadas mediante el asistente y que te permitirá llegar aún más lejos de las posibilidades iniciales de la app.

Launch Center Pro trabaja con URL-SCHEMES y lo muestra sin tapujos, esto, puede asustar a algunos, pero con el tiempo te ayudará a entender el funcionamiento y empezar a componer acciones mas complejas que las que permite el asistente de creación de acciones.

Launch Center Pro permite lanzar shortcuts y ejecutarlos directamente desde la app o desde las notificaciones, lo ahorra un paso a la hora de ejecutarlos.

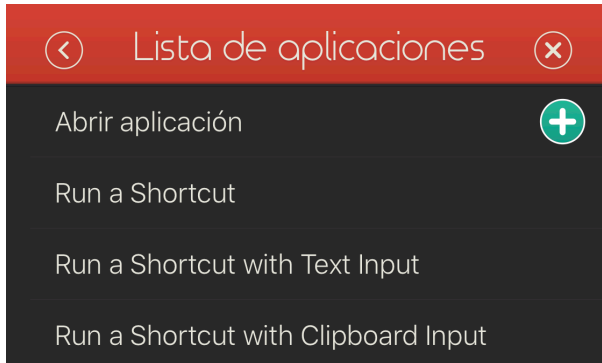
Ambas apps son en sí mismas tan complejas como se quiera llegar, aunque están mucho más limitadas que Shortcuts.

9.3.1. Shortcuts en Launcher

Para integrar un shortcut en **Launcher** debes pulsar sobre el botón de añadir nuevo en la pantalla principal, y en el menú de los diferentes tipos de launchers que hay elegir Launcher de aplicación.

A continuación, en la lista titulada *Se puede iniciar aplicación instalada* aparecerá Shortcuts por su nombre en inglés.

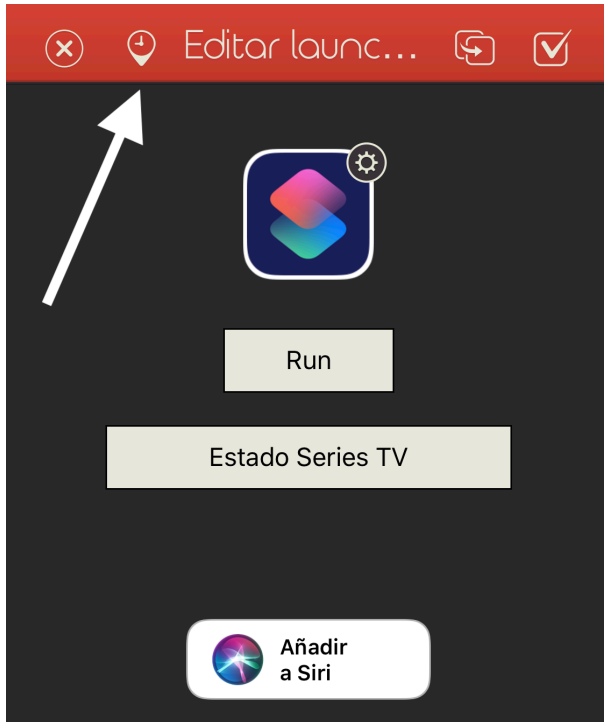
Las opciones que ofrece son: * Abrir aplicación * Run a Shortcut (Ejecutar un shortcut) * Run a Shortcut with Text Input (Ejecutar un shortcut con entrada de texto) * Run a Shortcut with Clipboard Input (Ejecutar un shortcut con entrada desde el portapapeles).



OPCIONES PARA TRABAJAR CON SHORTCUTS DESDE LAUNCHER

La opción de **abrir aplicación** no es más que abrir Shortcuts y que se muestre la galería para que elijas que shortcut ejecutar, por lo que no necesita más explicación.

Ejecutar un shortcut ejecuta un shortcut directamente al pulsar sobre el icono, para ello solo habrá que indicar el nombre exacto del shortcut, espacios incluidos dentro de la casilla Shortcut name. Donde pone Run puedes borrar ese texto y escribir uno que describa el shortcut que se va a ejecutar para que sea más fácil de identificar.



OPCIONES DE DISPARADORES

Tal y como puedes ver en la imagen anterior, para configurar los activadores hay que pulsar en el icono señalado con la flecha que no en vano, es una mezcla del icono de geolocalización omnipresente en las app de mapas y un reloj, haciendo alusión a los dos tipos de activadores automáticos que permite la app.

Así de fácil es configurar el lanzamiento de un shortcut en esta aplicación. Ahora, ese icono estará disponible para lanzar dicho shortcut desde la app de Launcher, desde el widget de la app, o si has configurado un disparador, al hacer clic en la notificación de la app se ejecutará el shortcut directamente,

por lo que es la forma más rápida de ejecutar un shortcut de forma programada que hay en la actualidad.

En el caso de seleccionar “Ejecutar un shortcut con entrada de texto” a la hora de crear el acceso directo aparece un campo extra donde se puede indicar el texto que Launcher le pasará al shortcut.

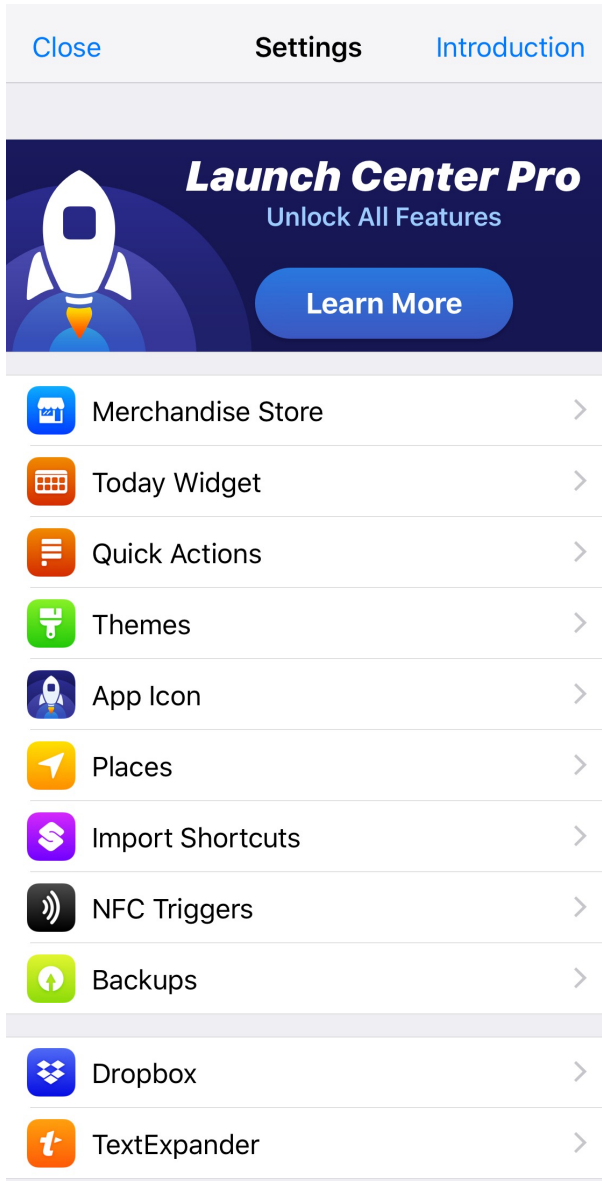
Si seleccionas “Ejecutar un shortcut con entrada del portapapeles” será Launcher el que obtendrá el texto que haya en ese momento en el portapapeles y se lo pasará al shortcut.

9.3.2. Shortcuts en Launch Center Pro

Integrar un shortcut en **Launch Center Pro** requiere un procedimiento ligeramente distinto y muy sencillo tras la última actualización en Diciembre de 2018.

La forma más fácil de añadir un shortcut a Launch Center Pro es mediante el shortcut que la propia aplicación es capaz de instalar en Shortcuts.

Para ello debes entrar en “Ajustes” en la rueda superior izquierda de la pantalla principal y pulsar sobre la opción “Import Shortcuts”.



PANTALLA DE CONFIGURACIÓN DE LAUNCH CENTER PRO

Dentro de ese menú tendrás la opción de instalar un shortcut en Shortcuts llamado **“Add Shortcuts To Launch Center Pro”** (Añadir shortcuts a Launch Center Pro).

Una vez instalado, desde Shortcuts, ejecutando el este shortcut te aparecerá la lista completa de los shortcuts que tienes instalados, selecciona los que quieras ejecutar desde Launch Center Pro, puedes seleccionar tantos como quieras. Esto te llevará a la pantalla de Launch Center Pro en modo editar, donde solo tendrás que seleccionar la posición de cada uno de los shortcuts para que se añada.

En primer lugar deberás pulsar el lápiz que hay arriba a la derecha de la aplicación para activar la edición de la pantalla actual y pulsar sobre el signo “+” que hay en el hueco en el que quieras colocar el acceso directo.

En el menú que aparece selecciona “Action” y a continuación “Action Composer”. Se desplegará una nueva pantalla donde puedes seleccionar donde puedes buscar Shortcuts en System Actions (más sencillo) o “All Supported Apps” (de forma manual) y hacer una búsqueda por la palabra “shortcuts”.

Si la seleccionas desde System Actions las opciones disponibles son:

- Launch Shortcuts: para abrir Shortcuts sin seleccionar ningún shortcut.
- Run Shortcut: tendrás la opción de añadir el nombre exacto de un shortcut para que se abra, si existe, al ejecutar la acción.
- Run Shortcut with Text: ejecuta un shortcut determinado y le envía un texto.
- Run Shortcut with Clipboard: ejecuta un shortcut determinado y le envía el contenido del portapapeles.
- Run Shortcut and Return: ejecuta un shortcut y cuando termines de hacer las tareas volverá automáticamente a Launch Center Pro.

- Run Shortcut w/Text and Return: ejecuta un shortcut enviándole un texto determinado y cuando termine de hacer las tareas volverá automáticamente a Launch Center Pro.
- Run Shortcut w/Clipboard and Return: ejecuta un shortcut con el contenido del portapapeles y cuando termine de hacer las tareas volverá automáticamente a Launch Center Pro.

Si eliges la opción de buscar en “All Supported Apps” verás que aparecen dos resultados:

- Shortcuts
- Shortcuts for Whatsapp

Pero son exactamente la misma opción, por lo que da igual la que selecciones.

Al terminar el “Action Composer” el apartado URL de la acción es shortcuts: que no es ni más ni menos que la URL-SCHEME para lanzar Shortcuts.

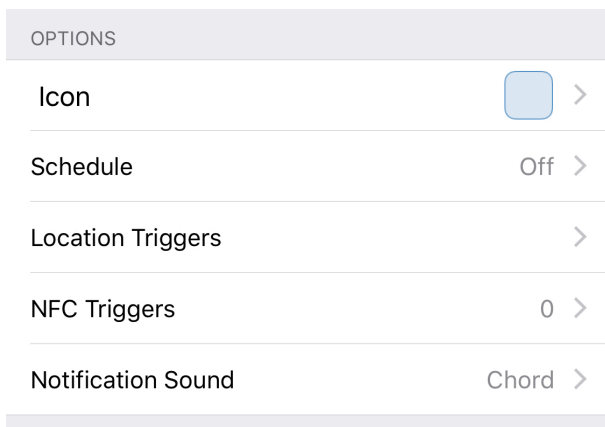
Si lo dejas así, esta acción solo abrirá la app Shortcuts. Debes conocer el esquema url (URL-SCHEME) de Shortcuts si quieres ejecutar un shortcut concreto:

shortcuts://run-shortcut?name=NombreExacto igual que en el caso anterior, la acción para ejecutar el shortcut Estado Series TV es: `shortcuts://run-shortcut?name=Estado%20Series%20TV` y si, al ser una URL no puede haber espacios, estos deben sustituirse por %20.

Para mandar al mismo tiempo un texto al shortcut que se vaya a ejecutar puedes añadir el parámetro input shortcuts://run-shortcut?name=Prueba&input=Este%20es%20mi%20texto de nuevo sustituyendo los espacios en el texto por %20.

Un poco más abajo de la URL en “Schedule” puedes configurar una hora determinada para que aparezca la notificación desde la que ejecutar el shortcut directamente. Y en “Location Triggers” el lugar donde te aparecerá la

notificación, si lo que quieres es ejecutar ese shortcut en un lugar determinado, incluso permitirá el uso de pegatinas NFC como lanzadores de shortcuts (mediante compra dentro de la app).



PANTALLA DE CONFIGURACIÓN DE LAUNCH CENTER PRO

9.4. DUE

La cuarta aplicación que se va a comentar es [DUE APP](#). Esta app permite recordar al usuario aquellas tareas que no pueden ser olvidadas. Básicamente es un sustituto de Recordatorios, la app nativa, pero con algunas funcionalidades extra que pueden ser un buen motivo de compra para algunos.

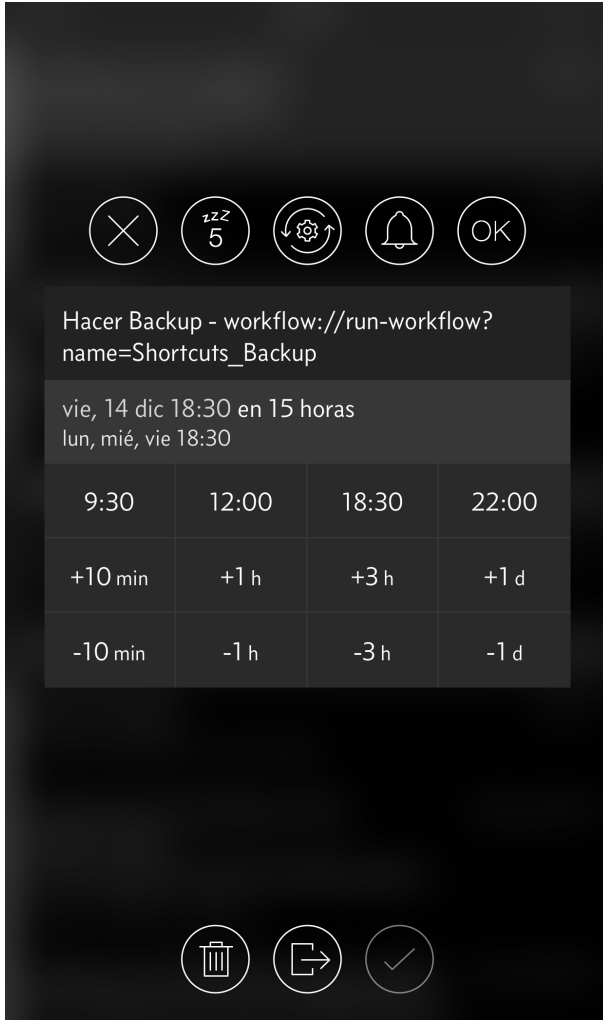
Estas otras funcionalidades pasan por una manera más ágil de gestionar/crear recordatorios repetitivos, lenguaje natural a la hora de crear esos avisos, y sobre todo el “Auto Snooze”, es decir, cuando venza una tarea, DUE irá repitiendo cada x minutos el aviso hasta que el usuario lo marque como completada, ya sea mediante [3D Touch](#), desde el Apple Watch o desde la App.

¿Qué relación tiene DUE con Shortcuts? Bien, mediante esquemas URL, se pueden configurar avisos que cuando aparezcan, tengan un formato de tal forma que al pulsar sobre ellos se ejecute un shortcut. Usando recordatorios se puede conseguir el mismo resultado, pero la manera de almacenar el atajo es algo menos sutil según se ha visto en el apartado 9.1.

La manera de crear un aviso en DUE:

1. Abrir la App, realizar un slide de arriba hacia abajo y se mostrará una pequeña caja de texto para poder crear el aviso.
2. Escribir un texto descriptivo, p.e. “Enviar mail” y a continuación: `shortcuts://run-shortcut?name=Nombre_del_shortcut`

El aviso tendrá este formato:



AVISO CON DUE APP

En cuanto a pros y contras, decir que DUE no permite establecer avisos según localización (para este tipo tienes Launch Center Pro), pero tiene a favor que es más “fácil” de

editar que crearlo con Siri, y es más “impertinente” para recordarte las tareas/acciones.

Sin duda DUE no es una app imprescindible para sobrevivir, pero la alta personalización en la creación de avisos repetitivos la hacen una candidata perfecta para los amantes del GTD.

Se puede enlazar shortcuts y enlaces con esquemas URL de cualquier aplicación soportada, por ejemplo [Things](#).

Para acabar este apartado, puedes descargar un shortcut que te facilitará la creación de recordatorios en DUE, lo puedes descargar [aquí](#). Se deja a manos del lector que lo modifique para añadir los bloques que sean necesarios para poder crear un recordatorio con repeticiones.

10. Puliendo y optimizando shortcuts

Shortcuts tiene muchísimos aspectos positivos, especialmente el hecho de poder crear tu mismo tus propias “apps”. En cuanto a sus carencias, destacar una, su falta de herramientas para depurar shortcuts.

Quizás la aplicación no estaba destinada a crear grandes shortcuts, pero el ingenio y la tenacidad de una comunidad ha provocado que día tras día surjan más y más utilidades creadas con Shortcuts.

En las páginas venideras podrás aprender algunas técnicas para poder depurar errores en tus desarrollos.

10.1. Compartir y preguntas de importación

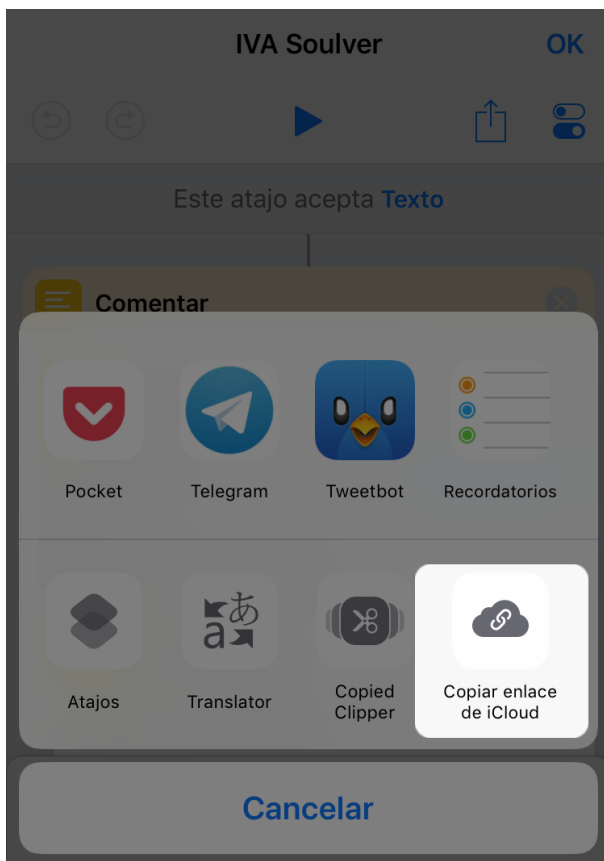
Uno de los motivos que contribuyeron al éxito de Workflow fue la comunidad de usuarios que se congregó entorno a la aplicación gracias a la posibilidad de compartir estos flujos de trabajo. Cómo no podía ser de otra forma, Shortcuts ha heredado esta característica que fomenta la colaboración y el enriquecimiento de la plataforma.

Desde la pantalla del *editor de shortcuts* es posible compartir un shortcut con otros usuarios.

Compartir como un archivo. Al seleccionar esta opción se genera un archivo de extensión *.shortcut que contiene toda la información del shortcut original. Este método resulta interesante para realizar una copia de seguridad de un shortcut.

Seleccionando **Copiar enlace de iCloud** a través del menú compartir se obtiene un enlace del tipo https://www.icloud.com/shortcuts/shortcut_id que se copia en el portapapeles. A continuación sólo hay que pegarlo en cualquier aplicación para compartir contenido, como Telegram

o Mail. Cuando el destinatario pulsa sobre el enlace se muestra una ventana dónde puede obtener una vista previa de las acciones, y descargarlo para añadirlo a su biblioteca pulsando “Obtener atajo”.



COPIAR ENLACE DE ICLOUD

Siempre que se comparte información con otros usuarios a través de internet es importante tener en cuenta qué información compartimos y cómo lo hacemos. Cuidar la privacidad y proteger nuestros datos en internet debería ser

una prioridad para cualquier usuario. En el mismo instante que se pulsa “compartir” esa información deja de pertenecernos y pasa a ser de dominio público en la red. Por este motivo es importante hacerse esta pregunta antes de compartir cualquier información personal en internet: *¿realmente quieres que esto lo conozca todo el mundo?*

Cuando se crea un shortcut es habitual incluir en él datos personales: direcciones, números de teléfono, claves o incluso contraseñas. Para evitar la publicación de este tipo de información sin que afecte al funcionamiento del shortcut, Shortcuts ha implementado las **preguntas de importación**.

Las preguntas de importación permiten eliminar la información confidencial de un campo determinado, cuando el usuario destinatario ejecuta el atajo debe responder aportando sus propios datos.

Para configurar las preguntas de importación puedes hacerlo a través del icono de *ajustes del editor de shortcuts*. A continuación, “Añadir nueva pregunta”, estas preguntas se lanzarán cuando el shortcut se importe en otro dispositivo, y las respuestas se guardarán en el flujo del shortcut “de destino”. Con cada pregunta se configura el valor de un parámetro dentro de la acción. Podemos utilizar tantas preguntas como creamos conveniente. Es importante tener en cuenta que los valores introducidos en las preguntas de importación sobrescriben la configuración anterior.

10.2. Debugging: Herramientas de ayuda

Esta es una de las facetas donde Shortcuts tiene un margen de mejora importante, lo sufrirás en cuanto empieces a manejarte con shortcuts de cierto calado. La app no tiene un “[debugger](#)”, es decir, no dispone de una herramienta específica para poder saber lo que está ocurriendo en la ejecución de un shortcut. Este aspecto, que parece menor cuando lo dedicas a shortcuts sencillos, cobra un valor más

relevante en cuanto te enfrentas con flujos más complejos que, por ejemplo, esperan resultados de REST APIs, de otros sub-shortcuts, etc...

Las técnicas tradicionales de debugging contemplan acciones como “breakpoints” (colocar paldas en puntos del código para que se detenga la ejecución en ese punto), inspección de variables (saber en cada momento el valor que tiene una variable, objeto, etc...), detener el código si se cumple una condición, y un largo etcétera de herramientas que ayudan a los desarrolladores a corregir (“debugar”) líneas de código y control de errores.

En Shortcuts todo es más “mecánico” y manual. Básicamente implica poner bloques de control (“*Mostrar Aviso*”, “*Vista Rápida*”, “*Visor de contenido*”) para poder inspeccionar el valor de salida de un bloque y saber si se está ejecutando todo como se espera. En algunos casos incluso bloques de acción como vibrar o vibrar dos veces seguidas sirven para saber rápidamente si se cumple una condición determinada o no.

Esta manera de trabajar es realmente tediosa, pero a día de hoy es la única manera que hay para detectar errores. La más usada es sin duda “*Mostrar Aviso*”, por que dispone del botón cancelar y si los valores que veas en esa inspección no son los esperados podrás parar para no continuar el resto del flujo. Es sin duda, una manera de ahorrar mucho tiempo de espera, ya que si en ese punto de ejecución no se obtienen los valores deseados, de nada sirve seguir ejecutando bloques.

De todas maneras no es malo usar “*Vista Rápida*” en numerosos puntos del shortcut cuando, por ejemplo, estás “navegando” dentro de un diccionario y quieres asegurar que estás en el nivel correcto para extraer los datos que necesitas en ese momento.

En futuras versiones de la App sería muy positivo que los desarrolladores se centraran en mejorar esta parte de la aplicación. La potencia de Shortcuts ya está fuera de toda

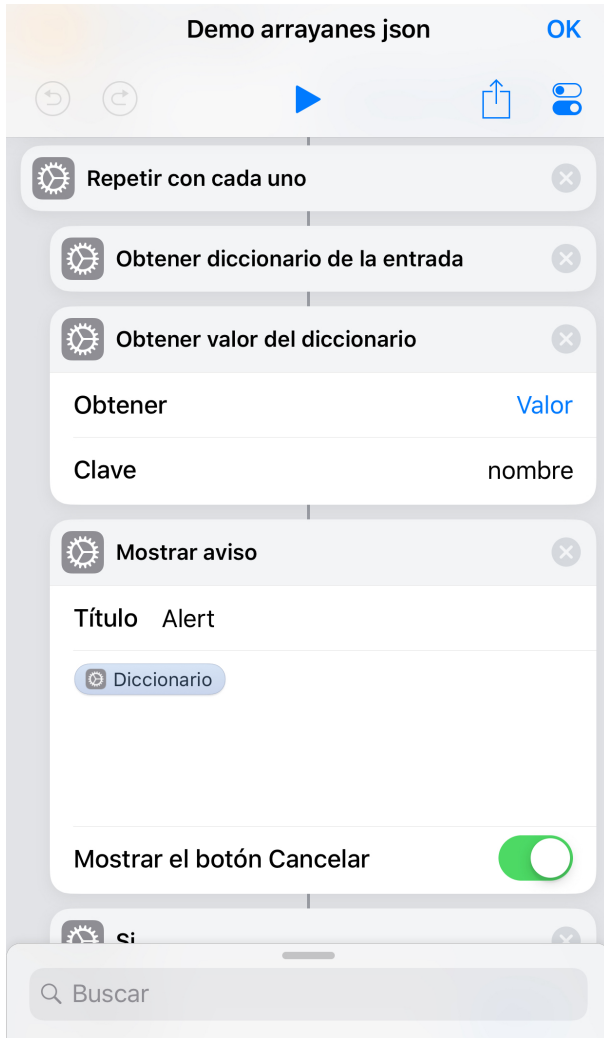
duda al poder integrarse con Web APIs, ahora se requieren bloques para poder controlar toda esta potencia y no quedarse en el “prueba-error” que en ocasiones puede llegar a ser muy frustrante, especialmente cuando empiezas a consultar APIs y no sabes a ciencia cierta lo que estás haciendo.

Como se ha dicho anteriormente, se hace indispensable un control de variables, y la posibilidad de poner una marca (flag) para detener el flujo de ejecución en un punto determinado y poder inspeccionar que valor tiene cada variable. En la captura puedes ver un “breakpoint” en Visual Studio .NET (herramienta de desarrollo de apps muy usada en entorno de desarrollo Windows)

En este capítulo aprenderás algunos trucos “Jedi” para poder corregir fallos y no perder la vida en ello.

10.2.1. Mostrar Aviso

Un clásico, es perfecto para “parar” el programa. La clásica parada (break) de cualquier depurador de errores (debugger) que te permite parar y ver que valor tienen las variables, con la pega que en Shortcuts es más “manual”. Aún así sigue siendo una buena opción, añadiendo uno o varios bloques en los puntos clave podrás ver qué va ocurriendo.



INTRODUCIENDO EL VALOR QUE SE QUIERE COMPROBAR EN EL TEXTO DE LA ACCIÓN SE PUEDE COMPROBAR EL VALOR QUE OSTENTA EN ESE MOMENTO DE LA EJECUCIÓN.

10.2.2. Copiar al portapapeles

Esta opción es muy válida cuando recuperes JSON, o HTML de una página, etc... Lo habitual es copiar estos resultados y tratarlos usando herramientas o servicios externos, ajenos a Shortcuts.

10.2.3. Vista Rápida

Vista rápida te ofrece una manera sencilla de ver el contenido que hay dentro de la acción inmediatamente anterior, ya sea imagen, texto, html, diccionario o lista. En ocasiones cuando no puede renderizar el contenido de forma vectorial lo pasteriza para que se vea correctamente.

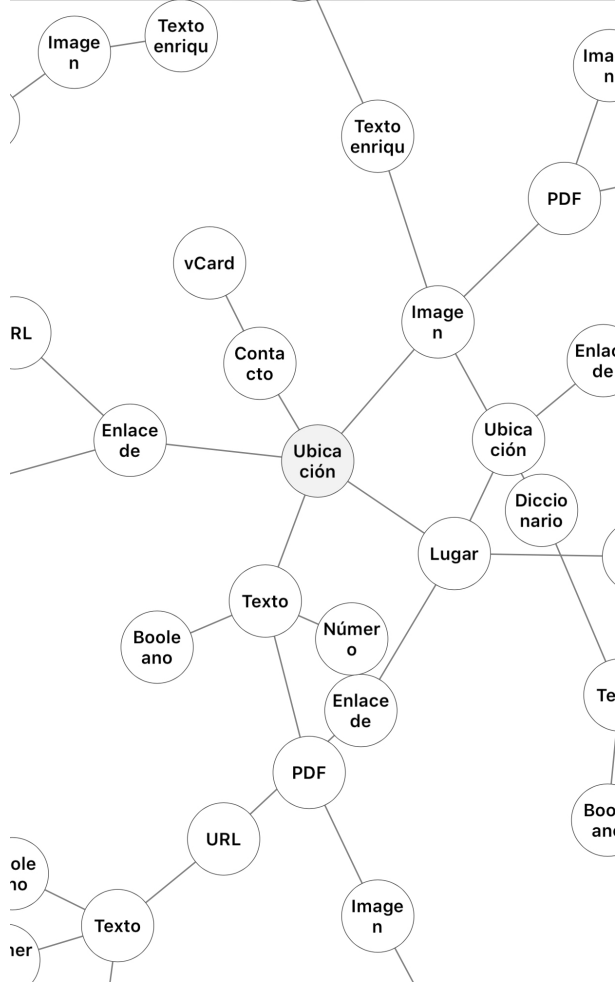
Es una de las herramientas para corregir errores mas usadas, si bien tiene la gran pega de no tener un botón de cancelar. Por lo que si te equivocas y usas esta acción dentro de un repetir con cada uno y este tiene muchas repeticiones te pasarás un buen rato dandole a "Ok".

Recuerda que abajo a la derecha, hay un botón que cambia la visualización a tabla cuando se trate de una lista o un diccionario, lo que facilitará enormemente la visualización y navegación por el contenido.

10.2.4. Visor de contenido

Este bloque permite ver en un momento de la ejecución, que tipo de objeto hay a la salida del bloque anterior, muestra las diferentes conexiones que hay entre los diferentes tipos de variable y como "razona" Shortcuts para facilitar al máximo esos cambios de "tipo de variable" que simplifican la vida al usuario. Si obtienes un texto pero éste es una URL que apunta a un PDF, podrás ver como el "Visor de Contenido" te muestra opciones para convertir ese texto en un PDF, ya que lee la URL y sabe que detrás hay un fichero en PDF.

“*Visor de contenido*” es muy útil para saber qué tipo de contenido está saliendo de una caja de acción determinada. Imagina un shortcut de tipo “extensión” (es decir, que estarás en una app y habrá que pulsar el botón de “compartir”) al que quieres que le llegue un mail de la app Spark y desees extraer los enlaces de ese mail. Para poder hacer eso, deberás asegurarte que lo que entra al shortcut es al menos Texto enriquecido o HTML, porque si llega texto plano no podrás extraer los enlaces al menos de una forma directa. Para casos como este es una buena idea usar el “*Visor de Contenido*”.



EL VISOR DE CONTENIDO MUESTRA UN DIAGRAMA PERT EN EL QUE ESTAN LAS ACCIONES Y LAS DEPENDENCIAS ENTRE ELLAS.

10.2.5. Uso de variables numéricas como Booleanas para separar código

Mediante un bloque “SI” y la variable especial “Entrada de Atajo” puedes hacer algo como esto:

⏪ ⏩ ▶ 📄 🔄

Si ✕

Entrada Contiene

Valor • Entrada de atajo

Si no ✕

Texto ✕

Valor inicial

Terminar si

Mostrar aviso ✕

Título Alert

Valor: ⚙️ Resultado de la acción Si

Mostrar el botón Cancelar

🔍 Buscar

SE COMPRUEBA SI EN LA ENTRADA DEL SHORTCUT HAY DATOS Y SI NO SE PASA EL TEXTO “VALOR INICIAL”

En el capítulo 2.3.2 se explicaron las diferentes opciones a la hora de tratar un shortcut como extensión. En ocasiones se dispone de shortcuts que pueden funcionar como autónomos o como extensión, por lo que la entrada del atajo puede venir ya con datos, o todo lo contrario, si lo ejecutas de forma autónoma no tendrá ningún valor a la entrada. En estos casos es útil usar esta fórmula, de este modo el atajo será funcional usándolo de un modo u otro.

10.2.6. Herramientas Externas

Es importante notar que para elaborar llamadas a Web APIs deberías usar [POSTMAN](#). Es una manera de ahorrar mucho tiempo en pruebas mediante ensayo-error que es la única manera de hacerlo directamente desde Shortcuts. Esta herramienta la has podido ver en el capítulo 7.

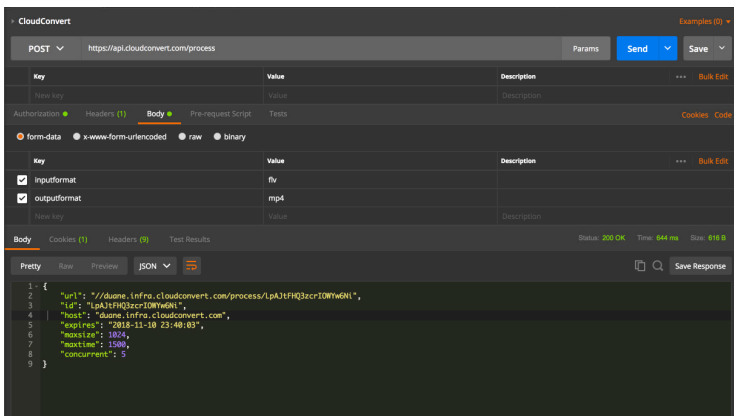
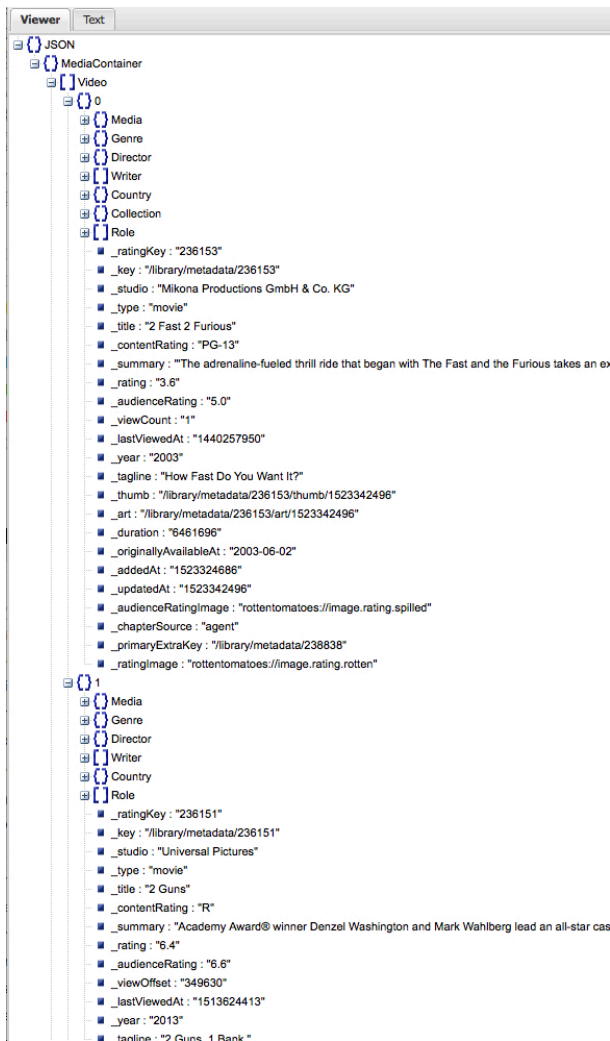


IMAGEN DE POSTMAN TRAS HABER HECHO UNA LLAMADA A UNA API MOSTRANDO LA RESPUESTA EN LA PARTE INFERIOR.

En cuanto a ficheros JSON, desde luego leer la estructura en texto sin ningún tipo de tabulación es mas bien tedioso, por lo

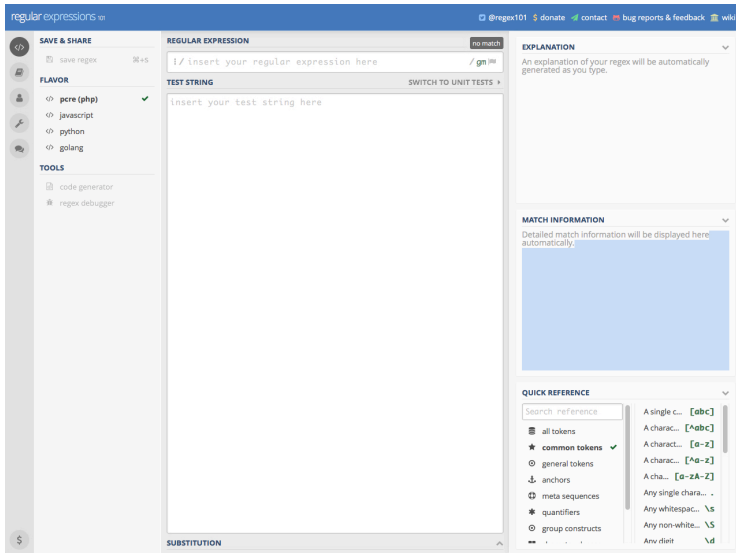
que usar herramientas online como [Jsonviewer](#) o [JSONLint](#) serán de gran ayuda para no perder la vista y la salud intentando recuperar algún valor.



JSON VIEWER TIENE LA CAPACIDAD DE MOSTRAR LOS JSON EN MODO ÁRBOL, POR LO QUE PODRÁS “CERRAR” AQUELLOS

SUBDICIONARIOS QUE NO TE INTERESEN PARA TENER UNA VISTA MAS LIMPIA DEL CONTENIDO.

Finalmente, para expresiones regulares, sin duda, [Regex101](#) es una gran herramienta, potente, didáctica (te explica lo que hace cada expresión que es escrita, posee una referencia rápida en formato ayuda...)



CAPTURA DE PANTALLA DE REGEX101.

11. Shortcuts y Salud

A menudo, los datos de salud se populan mediante los sensores del iPhone, Apple Watch y otras pulseras cuantificadoras o wearables (dispositivos electrónicos vestibles). Otras veces, es mediante apps, como Waterminder que controla tu ingesta de agua diaria, o MyFitnessPal para introducir las comidas que haces y llevar el control de los nutrientes.

Otra de las fuentes es Shortcuts, que, por ejemplo te puede ayudar a ir introduciendo tus datos de peso cada vez que te peses en tu báscula aunque ésta no sea inteligente.

Las acciones relativas a salud presentes en shortcuts, tal y como se listaban en el capítulo 3 son las siguientes:

- **Buscar muestras médicas**

- Entrada: ninguna.
- Parámetros:
 - Tipo de muestra
 - Fecha de las muestras
 - Filtros
 - Unidad: contar, porcentaje
 - Agrupar por: ninguno, minuto, hora, día, semana, mes, 3 meses, año
 - Ordenar por: valor, fecha de inicio, fecha de fin, duración, origen, nombre, aleatorio
 - Límite
- Salida: los valores de las muestras que coincidan con los parámetros dados.

- **Obtener detalles de muestra médica**

- Entrada: Muestra médica
- Parámetros:
 - Obtener: tipo, valor, unidad fecha de inicio, fecha de fin, duración, origen, nombre

- Salida: Valor obtenido de la muestra médica de la entrada.
- **Registrar muestra médica**
 - Entrada: muestra medica.
 - Parámetros:
 - Tipo: Cualquiera de las disponibles en la App Salud.
 - Fecha
 - Salida: Añadirá a la app Salud la muestra que se haya especificado
- **Registrar entreno**
 - Entrada: ninguna.
 - Parámetros:
 - Tipo: Cualquiera de las disponibles en la App Salud.
 - Fecha
 - Duración
 - Calorías
 - Distanciar (es una errata, se refiere a distancia)
 - Salida: Añadirá el entrenamiento seleccionado junto con sus datos a la App Salud.

El tipo de muestras que pueden registrarse son las siguientes:

- Actividad electrodérmica
- Actividad sexual
- Administración de insulina
- Agua
- Altura
- Análisis de sueño
- Atención plena
- Azúcar dietético
- Ácido pantoténico
- Biotina

- Brazadas
- Cafeína
- Calcio
- Calidad del moco cervical
- Capacidad vital forzada
- Carbohidratos
- Cloruro
- Cobre
- Colesterol alimentario
- Contorno de la cintura
- Cromo
- Distancia andando/corriendo
- Distancia en bicicleta
- Distancia en silla de ruedas
- Distancia nadada
- Energía alimentaria
- Energía en actividad
- Energía en reposo
- FC en reposo
- Fibra
- Flujo espiratorio máximo
- Folato
- Fósforo
- Frecuencia cardiaca
- Frecuencia respiratoria
- Glucosa en sangre
- Grasas monoinsaturadas
- Grasas poliinsaturadas
- Grasas saturadas
- Grasas totales
- Hierro
- Impulsos
- Índice UV
- Índice de masa corporal
- Índice de perfusión periférica
- Magnesio
- Manchado

- Manganeso
- Masa corporal sin grasa
- Menstruación
- Molibdeno
- Niacina
- Nivel de alcohol en sangre
- Número de caídas
- Pasos
- Peso
- Pisos subidos
- Porcentaje de grasa corporal
- Potasio
- Proteínas
- Pruebas de ovulación
- Riboflavina
- Saturación de oxígeno
- Selenio
- Sodio
- Temperatura corporal
- Temperatura corporal basal
- Tensión arterial
- Tiamina
- Uso de inhaladores
- VO₂ máx.
- Variabilidad de la FC (Frecuencia cardiaca)
- Vitamina A
- Vitamina B12
- Vitamina B6
- Vitamina C
- Vitamina D
- Vitamina E
- Vitamina K
- Volumen espiratorio forzado, 1 s
- Yodo
- Zinc

Puedes observar como la lista de muestras que se pueden agregar es muy extensa, por lo que seguramente cubrirá tus necesidades de una manera u otra.

El funcionamiento es parecido al de otros tipos de información, como los objetos de iTunes o Apple Music con la posibilidad de filtrar, ordenar y seleccionar el número de resultados que se obtendrán.

12. Ejemplos prácticos

Con lo que has aprendido en este libro ahora vas a poder practicar haciendo unos cuantos ejemplos que unan los conocimientos adquiridos en distintos capítulos para multiplicar exponencialmente el potencial de tu aprendizaje en Shortcuts.

Se ha dividido en dos tipos:

- **Para aprender:** son ejemplos destinados a solucionar problemas complejos y a la vez mecánicos que surgen a la hora de hacer shortcuts elaborados.
- **Ejemplos útiles:** son ejemplos especialmente pensados para poner en funcionamiento todo lo aprendido y que una vez terminado tengas un shortcut plenamente funcional que solucione una automatización que pueda serte de utilidad.

12.1. Para aprender

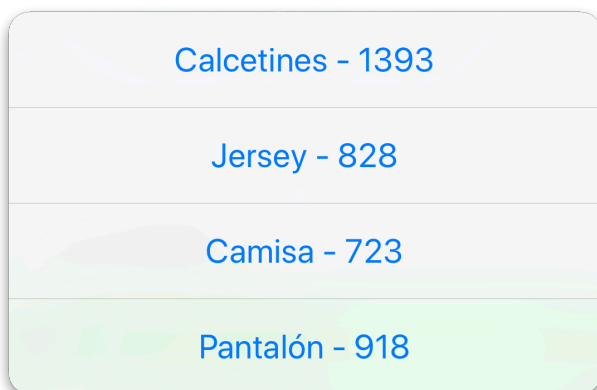
12.1.1. Emparejar elementos de dos listas

A menudo, cuando trabajas con datos te encontrarás con que tienes elementos de dos listas diferentes que necesitas unir bien para crear un diccionario clave:valor o bien para que la lista esté más completa. Como por ejemplo:

Una lista en la que tienes los objetos mas comprados en una tienda y en otra lista, en el mismo orden, las unidades vendidas de dicho objeto:

- Calcetines, Jersey, Camisa, Pantalón
- 1393, 828, 723, 918

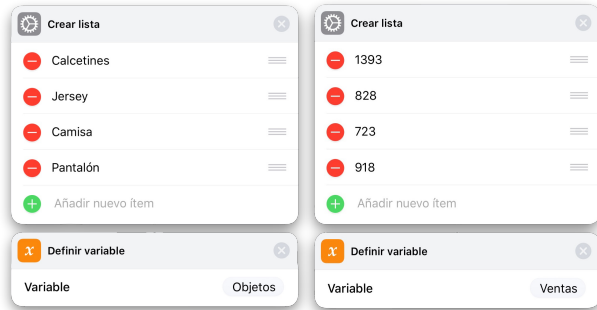
Vamos a ver el primer caso en el que lo que vas a conseguir es una lista de este tipo:



Calcetines - 1393
Jersey - 828
Camisa - 723
Pantalón - 918

RESULTADO QUE OBTENDRÁS

Normalmente estos datos los obtendrás de un servicio web o de una base de datos que hayas creado. En este caso, para ejemplo se crearan dos listas, la primera almacenada en una variable llamada “Objetos” y la segunda en una variable llamada “Ventas”.



LAS LISTAS Y LAS VARIABLES A LAS QUE SE ASIGNAN

Para emparejar los elementos de la lista tendrás que elegir una de ellas para que lidere, por así decirlo, el emparejamiento. En este caso usarás la lista de los artículos, por tanto habrá que recuperar el valor de la variable que la contiene mediante una acción Obtener variable. A continuación se usará una acción del tipo Repetir con cada uno para que Shortcuts reitere la acción con cada uno de los elementos de la lista.

Dentro de las acciones Repetir o Repetir con cada uno hay dos variables intrínsecas que son muy útiles: * **Repetir índice**: es el número de repetición en la que se encuentra la ejecución, esta variable tendrá tantos valores como objetos haya dentro de la variable principal comenzando por 1. Si la variable sobre la que se ejecuta la repetición tiene 4 objetos, como en este caso, repetir índice ira incrementando su valor y será 1, 2, 3 y 4 sucesivamente. * **Repetir ítem**: es el valor concreto sobre el que se esta llevando a cabo la repetición. En este caso, la primera vez este valor será “Calcetines”, en la segunda iteración será “Jersey” y así sucesivamente.

Dentro de la acción Repetir con cada uno recuperaremos la variable Ventas, y para seleccionar solo el valor que corresponda con cada repetición usarás la acción Obtener ítem de la lista y en el valor “Obtener” seleccionaremos Ítem

del índice para que aparezca la casilla “Valor” dónde se usará la variable interna Repetir índice que se acaba de explicar.

De esa manera en la primera repetición el valor que ofrecerá será 1393. En esta primera vuelta, el valor de Repetir ítem como puedes deducir será el primer elemento de la lista “Objetos”, es decir, Calcetines.

Por tanto, solo quedará recuperar en una caja de texto ambos valores con el formato que más te guste

Aprovecharemos, por tanto el valor repetir índice para obtener sobre una acción de texto el Objeto que corresponde con la repetición en la que estemos.

The image displays a vertical stack of workflow configuration panels:

- Repetir con cada uno**: The main action panel, featuring a gear icon and a close button.
- Obtener variable**: A panel with an orange 'x' icon. The 'Variable' field is set to 'Ventas'.
- Obtener ítem de la lista**: A panel with a gear icon. The 'Obtener' field is set to 'Ítem del índice' and the 'Índice' field is set to 'Repetir índice'.
- Texto**: A panel with a yellow menu icon. It contains a sequence of actions: 'Repetir ítem' followed by 'Ítem de la lista'.
- Terminar repetición**: A final panel indicating the end of the loop.

Below the configuration panels is a preview window showing the output of the workflow:

```
Calcetines - 1393
```

The preview window includes standard UI controls: a zoom-in/out icon, a three-dot menu, and a share icon.

ACCIÓN REPETIR CON CADA UNO CONFIGURADA

Puedes ver, abajo del todo que la salida es una lista en la que ya están correctamente emparejados los objetos y sus unidades vendidas. Ya solo queda usar una acción “Seleccionar de la lista” para que se muestre de una manera más visual.

Puedes descargar el ejemplo terminado por si has tenido alguna duda.



[DESCÁRGALO AQUÍ](#)

12.1.2. Ordenar un diccionario

En el capítulo de Diccionarios se veía como éstos son listas de pares “clave:valor” desordenadas con una única excepción, que se hubieran introducido los datos mediante el texto clave=valor unidos por &.

Va a ver, mediante un ejemplo guiado cómo convertir cualquier diccionario, sea cual sea su origen, en un diccionario ordenado según el criterio que veas más oportuno. También se verán sus limitaciones.

Esto es especialmente útil si quieres mostrar una lista enriquecida que tenga sus elementos por orden alfabético, por ejemplo.

Comenzaremos usando un diccionario en formato texto usando el formato JSON para que puedas copiarlo y pegarlo en tu shortcut:

```
{  
    "1": "a",  
    "2": "b",  
    "3": "c",  
    "4": "d"  
}
```

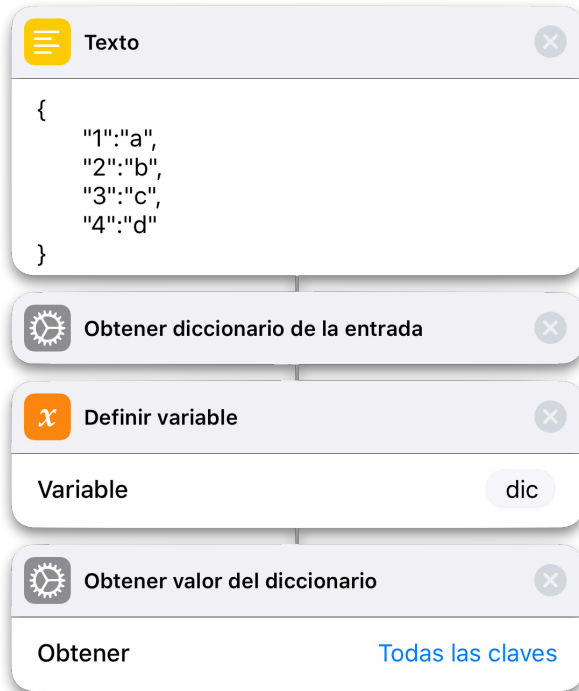
Al introducir el texto un una acción de “Texto” y usar una acción “Obtener diccionario de la entrada” y ejecutar el shortcut verás que el resultado es el mismo JSON desordenado:



CUANDO EL TEXTO SE CONVIERTE EN UN DICCIONARIO ESTE SE RECUPERA DESORDENADO.

Esto no quiere decir que lo desordene antes de guardarlo por gusto y que ya siempre esté en ese orden que ves en la imagen. De hecho, lo mas probable es que a ti te aparezca con otro orden completamente distinto, e incluso que el orden cambie con cada ejecución.

Para continuar, ahora guarda el diccionario en una variable, a la que llamarás dic para poder recuperarla varias veces. Y usando una acción “Obtener valor del diccionario”, en el campo “Obtener” cambia “Valor” por “Todas las claves”.



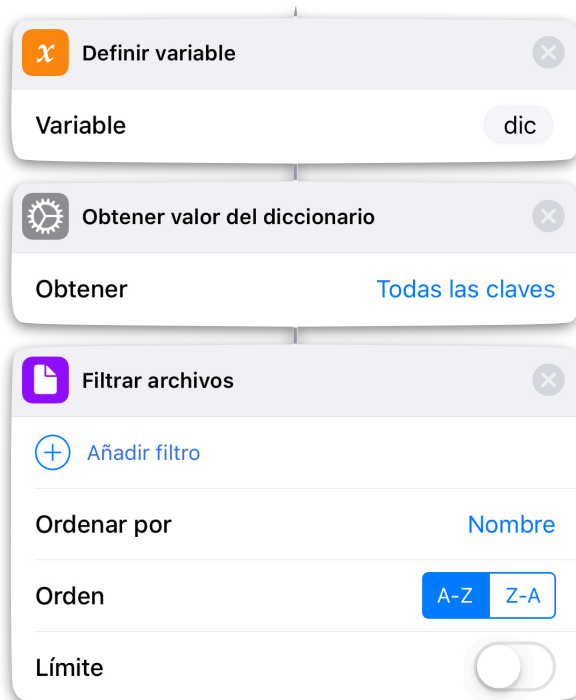
SE GUARDA EL DICCIONARIO EN UNA VARIABLE Y SE OBTIENEN TODAS SUS CLAVES.

Con esto se obtiene una lista de todas las claves que hay dentro del diccionario, que seguirá el mismo desorden que posee el diccionario.

Para ordenar esa lista, usaremos una acción que es demasiado poco evidente para el funcionamiento al que nos tiene acostumbrados apple. Se trata de la acción “Filtrar archivos”, que, aunque en la descripción pone que sirve para ordenar o filtrar archivos de un tipo determinado, también filtra y ordena listas de datos.

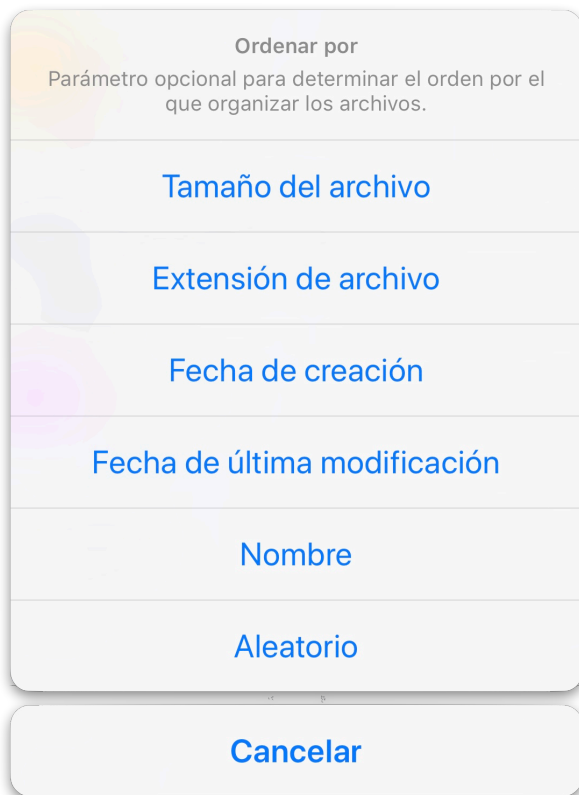
En este caso no es necesario filtrar, solo ordenar, por lo que solo configurarás el campo “Ordenar por” cambiando el valor

“Ninguno” por “Nombre”. Y en orden “A-Z” para que sea descendente.



CONFIGURACIÓN DEL FILTRO Y ORDEN DE LA LISTA.

Se comentó al comienzo del ejemplo que ordenar un diccionario tenía algunas limitaciones, pues bien, las limitaciones vienen dadas por las opciones disponibles en la acción “Filtrar archivos” que son las de la siguiente imagen:



CONFIGURACIÓN DEL FILTRO Y ORDEN DE LA LISTA.

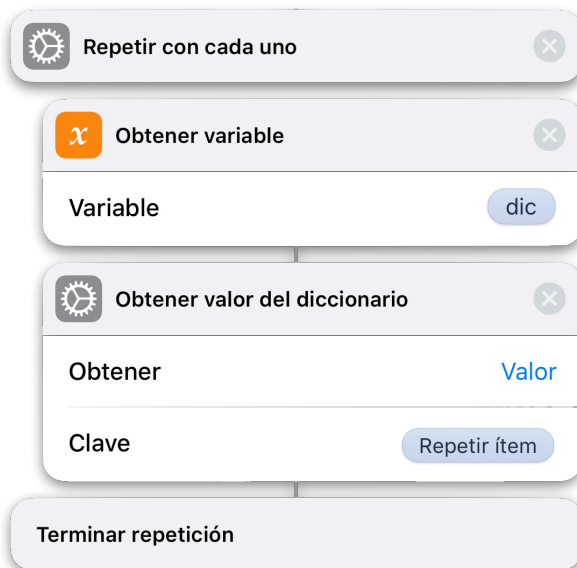
Como puedes observar, está preparado para trabajar con archivos, por lo que la única opción que sirve para ordenar una lista es la de "Nombre".

Una vez se selecciona esta opción se puede seleccionar, mediante el campo "Orden" hacer que éste sea ascendente seleccionando "A-Z" (1,2,a,b,...) o descendente seleccionado "Z-A" (b,a,2,1...).

Después de este inciso, ahora tienes una lista ordenada de todas las claves del diccionario, pero claro, solo las claves, los valores se quedaron atrás.

Ahora necesitamos recuperar los valores usando las claves que si están ordenadas, para ello se usará una acción “Repetir con cada uno” en el que se reiterará por los elementos la lista ordenada que se usarán para recuperar sus valores dentro del diccionario.

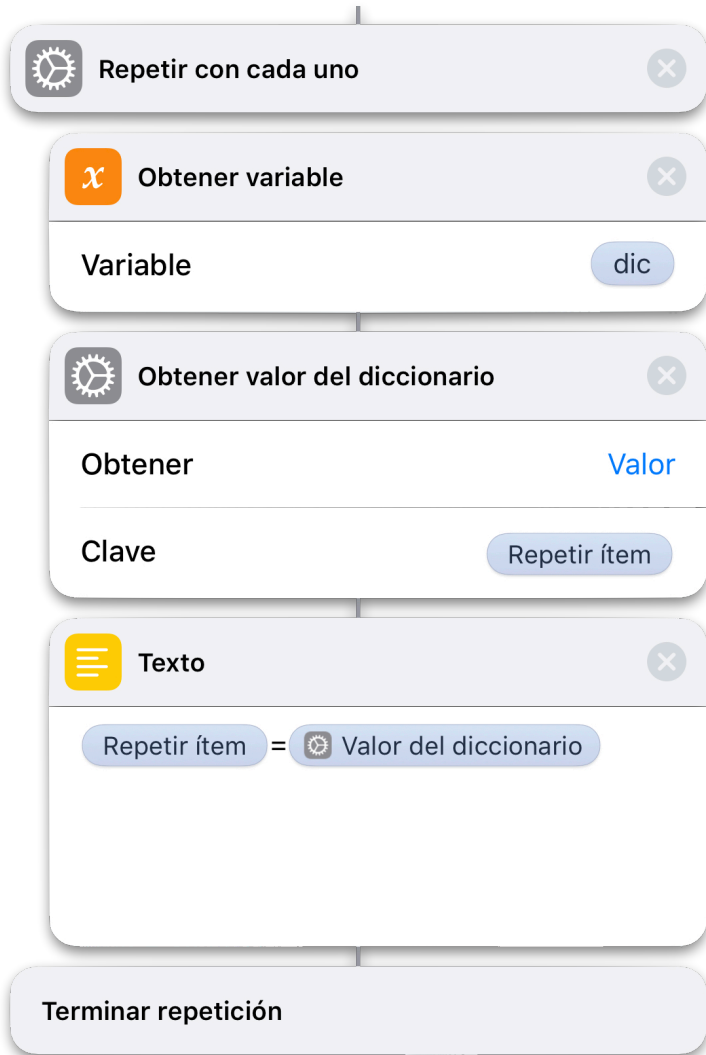
Lo primero que se debe hacer dentro de la acción de “Repetir con cada uno” es recuperar el diccionario y con la acción “Obtener valor del diccionario” la configurarás para que la clave el “Repetir ítem” de la siguiente manera:



ACCIÓN DE REPETIR SIN TERMINAR.

El resultado de “Repetir ítem” es la clave del diccionario que toque en la repetición. Y El resultado de “Obtener valor del diccionario” con esta configuración es la “clave” que le

corresponde en el diccionario, por lo que ahora se pueden unir de forma ordenada en un texto con el formato clave:valor unidos por &.



ACCIÓN DE REPETIR COMPLETAMENTE TERMINADA

Ahora solo queda combinar la lista que se obtiene al terminar la repetición con una acción “Combinar texto” usando como “Separador” uno personalizado donde se escribirá una &.

De nuevo se usará la acción “Obtener diccionario de la entrada” para convertir el texto en un diccionario. Y si añades una acción “Seleccionar de la lista” podrás ver como, ahora si, se muestran en perfecto orden alfabético.



SE VUELVEN A UNIR LOS PARES Y SE CREA EL DICCIONARIO DE NUEVO.

Puedes descargar el shortcut terminado



DESCÁRGALO AQUÍ

11.1.3 Enviar mensaje por Whatsapp con el tiempo de llegada.

En este ejemplo se van a usar dos shortcuts trabajando juntos y trabajo con diccionarios. Para ello deberás tener:

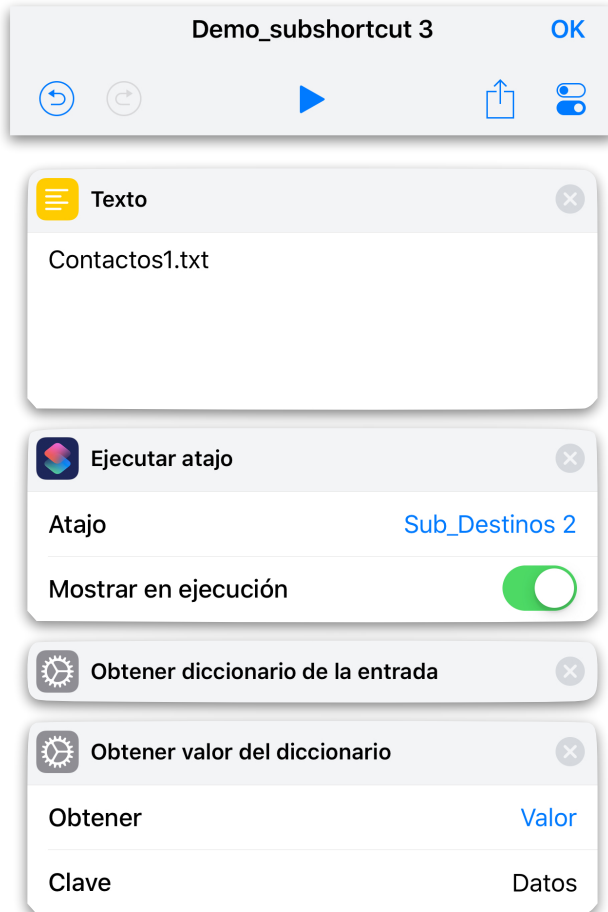
1. Un fichero de configuración más complejo, formato JSON con diccionarios anidados.(uno dentro de otro)
2. Un shortcut principal que pasará el fichero de configuración a recuperar a un subshortcut. [Descarga](#)
3. El subshortcut devolverá los datos (en forma de JSON). [Descarga](#)
4. De vuelta al shortcut principal, se usará el bloque *Repetir con cada uno* dándole como entrada un a lista y se recorrerá obteniendo datos para después crear una lista de selección.
5. Finalmente, el usuario selecciona un ítem de la lista y el shortcut le enviaría un mensaje por WhatsApp indicando el tiempo que tardará en llegar.

En primer lugar el fichero de configuración:

```
{
  "Datos":
    {
      "mama":
        {
          "telefono": "666778899",
          "direccion":
            "su+direccion+de+casa"
        },
      "trabajo":
        {
          "telefono": "111223344",
```

```
        "direccion": "el+trabajo"
    },
    "amigo":
    {
        "telefono": "99900999",
        "direccion": "direccion+amigo"
    }
}
```

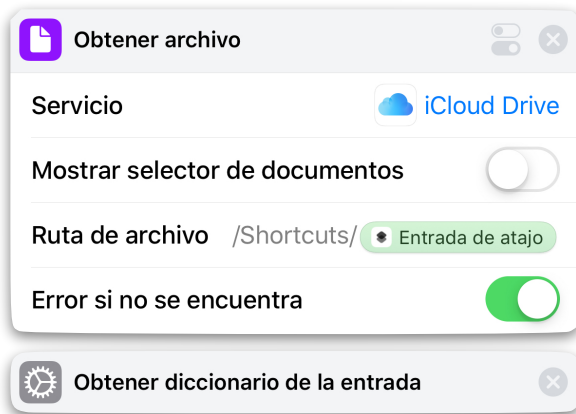
Este debe ir salvado en iCloud > Shortcuts con el nombre: Contactos1.txt. En el shortcut principal se pasa al subshortcut el fichero que debe recuperar:



EN EL BLOQUE TEXTO SE PASA EL NOMBRE DEL FICHERO

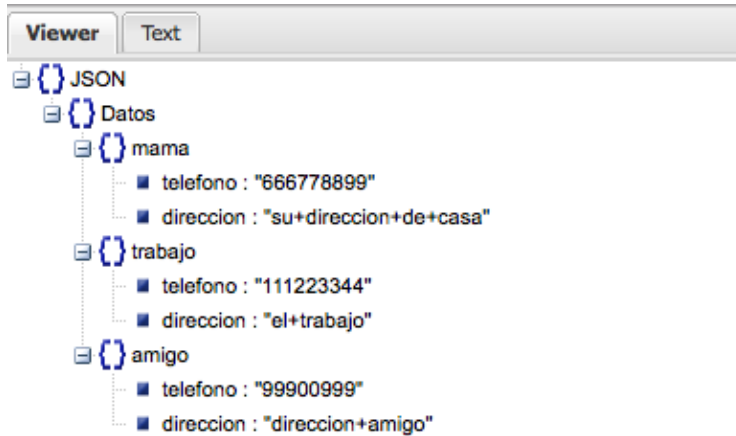
Recuerda que debes enlazar el bloque “*Ejecutar atajo*” al shortcut llamado “Sub_Destinos 2”, el subshortcut.

En el subshortcut se recibe el nombre del fichero, usando la variable especial “Entrada de Atajo”, lo descarga, y lo devuelve:



SUBSHORTCUT RECIBE EN LA VARIABLE ESPECIAL EL NOMBRE DEL FICHERO

Ya de vuelta al shortcut principal, se recibe un diccionario y se entra al su primer nivel:



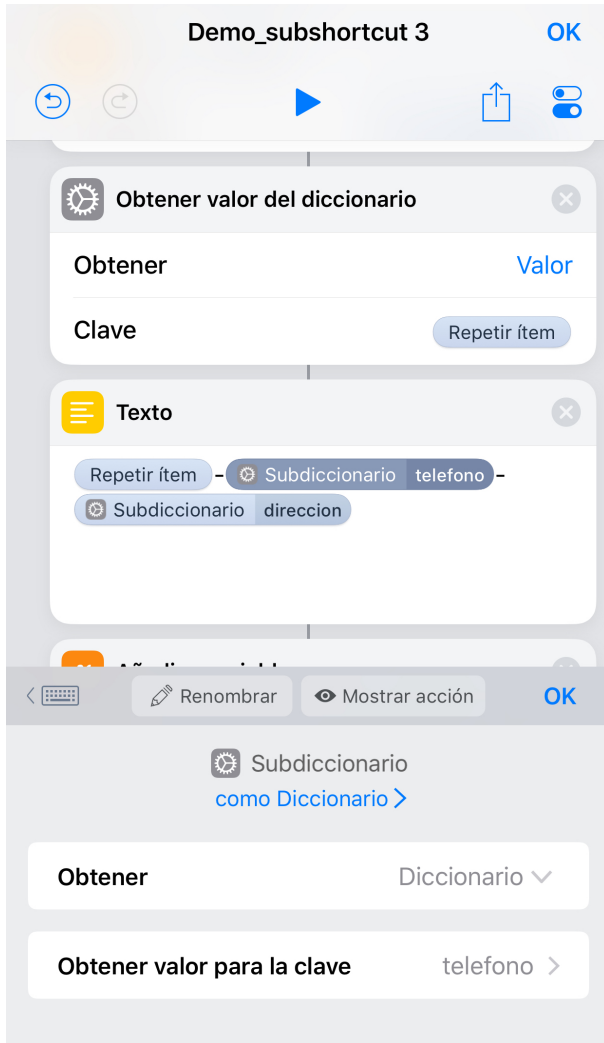
DICCIONARIOS ANIDADOS

A continuación se recuperan las claves, para poder obtener una Lista, y esta usarla en el bloque *“Repetir con cada uno”*. En el bucle. Para recuperar datos se necesita en primer lugar recuperar el diccionario *“Obtener variable”*.



REPETIR CON CADA UNO DANDO UNA LISTA DE ENTRADA

Se recupera el “subdiccionario” que lleva el nombre del ítem de la lista recorrida: mama, trabajo y amigo y, usando variables mágicas, con un tipo Diccionario se pueden recuperar los valores de “teléfono” y “dirección”.



CAMBIO DE TIPO DE LA VARIABLE MÁGICA COMO DICCIONARIO

Finalmente, en una acción “*Texto*” se conjugan estos valores, y se añaden a la variable “Valores”.

La parte final del shortcut es para darle algún sentido a todo esto, y simplemente deja al usuario seleccionar un elemento, lo divide para obtener los datos, y genera un WhatsApp indicando al destinatario el tiempo que va a tardar en llegar.

12.2. Ejemplos útiles

En este último capítulo se verán unos ejemplos más elaborados de shortcuts que pondrán a prueba parte de lo aprendido a lo largo del libro para que veas como la combinación de todo puede dar lugar a shortcuts realmente interesantes.

No cabe duda que el shortcut más importante es aquel que resuelva una necesidad que tengas, por tanto, el potencial se desarrollará una vez hayas empezado a crear los tuyos propios.

Como has llegado ya a este punto, en estos ejemplos no se va a ir detallando con imágenes todo lo necesario para crear los shortcuts, se irá escribiendo como si de una receta se tratara, pero se dará por hecho que se ha comprendido el contenido del libro, en sus aspectos más básicos para ser mas eficientes y dedicar más tiempo a explicar combinaciones complejas que pueden resultarte nuevas y más difíciles.

Al final de cada ejemplo encontrarás el enlace de descarga del shortcut tal y como se ha explicado en el ejemplo, así podrás instalarlo y ver si hay algo que hayas hecho mal para autocorregirte.

Si hay una sección que destaca sobre el resto en cuanto a novedades en Shortcuts 2.2 esta es sin duda la referida a notas. Se han incluido 3 acciones, y modificado una cuarta que ya existía:

- "Crear Nota"
- "Buscar Notas"

- “Ver Nota”
- “Añadir a Nota”

"**Crear Nota**" ya estaba presente anteriormente, pero en esta acción se ha incluido una opción llamada "Mostrar la Hoja de Redacción" de manera que si se desmarca se pueden ya crear notas en background, es decir, sin que se abra en primer plano la aplicación de Notas. Eso sí, la nota se creará en la raíz de Notas, no se permite definir dónde guardar la nota creada.

Las notas creadas de este modo deben ser en modo texto plano, lo cual las hace todavía muy limitadas y lejos de apps como Bear (ver ejemplo en el capítulo 3.4.3), cuya interacción con Shortcuts a día de hoy es mucho mayor.

"**Buscar Notas**": Este bloque incluye diferentes opciones que facilitan el filtrado de notas. Se pueden filtrar notas según "**cuerpo**" (texto de la nota), "**fecha de creación**", "**última modificación**" y "**nombre**" (título de la nota). Todos estos filtros pueden ser conjugados entre sí, es decir, se pueden filtrar notas que contengan "perro" y que su "fecha de creación" "es anterior al" una fecha determinada.

Las notas filtradas se pueden ordenar por:

1. Cuerpo
2. Carpeta
3. Fecha de Creación
4. Última modificación
5. Nombre
6. Aleatorio

A su vez, este orden es por antigüedad de forma ascendente/descendente.

Finalmente, comentar que las notas devueltas se pueden limitar en número mediante la última de las opciones de

configuración de este bloque, por ejemplo limitando a 6 notas el resultado del filtro.

Una última consideración al respecto de esta acción. En el momento de escribir estas líneas debe ir precedida del bloque "**Nada**". El motivo es un bug de programa, de manera que si a la entrada de "Buscar Nota" hay algún dato/variable, la búsqueda no retorna nada.

"**Ver Nota**": Esta acción recibe una nota como entrada y abre la app de Notas con la nota.

"**Añadir a Nota**": Este bloque como su propio nombre indica te permitirá añadir texto (en formato texto plano) a una nota existente. La manera de utilizar este bloque es asignar a la entrada del bloque el texto a añadir, y en la opción "nota", la variable que contiene la nota ya existente que va a ser modificada.

Para describir todos estos bloques puedes descargar este ejemplo:



[Descárgalo aquí](#)

12.2.1. Crear un blog usando Telegra.ph

Los creadores de [Telegram](#) crearon [Telegra.ph](#) en 2016. Esta plataforma te permite crear un artículo de la forma más minimalista que existe.

Title

Your name

Your story...

PUBLISH

ESTO ES TODO LO QUE VERÁS AL INGRESAR EN TELEGRA.PH

Sin embargo, no creas que por no tener botones no tiene opciones, es una plataforma muy potente. Pero su filosofía hace que no haya una página principal donde ir recopilando tus artículos, por lo que siempre deberás guardar el enlace si no quieres perderlos o create una cuenta con el BOT de Telegra.ph en Telegram.

En este ejemplo vamos a crear un shortcut relativamente complejo en el que tendremos la posibilidad de imitar el funcionamiento de un blog usando la plataforma Telegra.ph.

La estructura del shortcut es la siguiente:

- Diccionario de configuración
- Comprobación de si es la primera vez que se ejecuta. Si es la primera vez:
 - Se crea la configuración

- Se crea la página principal del blog
- Se crea un menú:
 - Crear entrada
 - Se crea una página nueva de Telegraph
 - Se guarda un resumen de lo escrito y su url en la página principal.
 - Ver blog

11.2.1.1. Diccionario de configuración

Se empieza con un diccionario de configuración en el que son necesarios los siguientes campos para dar de alta tu usuario en la plataforma de Telegra.ph y que te permitan usar su API.

Esto, por supuesto, es gratuito, así que no debes preocuparte por nada.

El diccionario tendrá las siguientes claves y valores de ejemplo. Te aconsejo que uses estos y cuando lo hayas terminado configures tus datos:

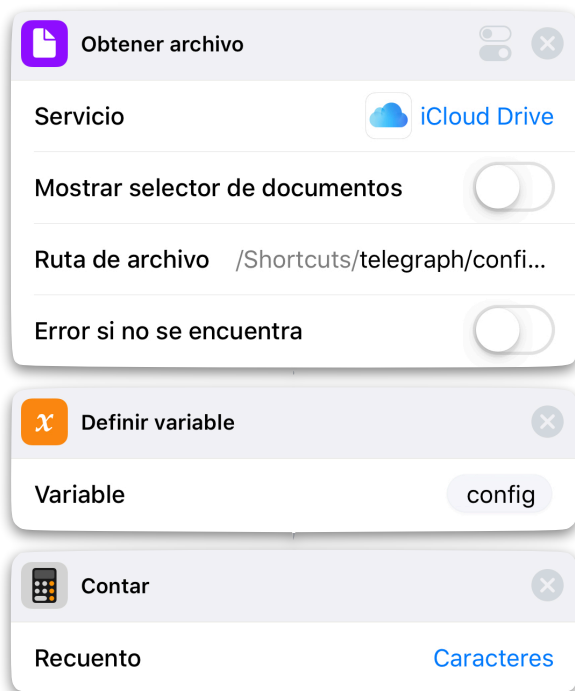
- titulo: Mi super blog
- autor: Miguel de Cervantes
- nAutorCorto: mcervantes

En este shortcut se va a crear una carpeta en iCloud que vaya guardando la configuración y archivos necesarios para que todo vaya bien, así que lo primero es generar un método para comprobar si es la primera vez que se usa el shortcut y por tanto hay que crear toda la estructura de carpetas y archivos o ya se ha usado y se debe leer la configuración que ya se hizo anteriormente.

Para ello usa una acción “Obtener archivo”, desactiva la opción de “Mostrar selector de documentos” y en la ruta del archivo escribe telegraph/config.json. Desactiva también la casilla de “Error si no se encuentra”, ya que ese error lo manejaremos nosotros.

12.2.1.2. Comprobación de existencia de configuración

Se guarda el archivo en una variable que se crea para ello a la que se llamará “config”. De cara a poder ver “si” existe el archivo o no, se cuentan los elementos que contiene la variable tras recuperar el archivo. Como es lógico, si tiene 0 caracteres es que el archivo no existía por lo que no hay nada escrito en el y si tiene más de 1 carácter en su interior es que sí existe. Podría haberse hecho contando ítems.



BUSCA EL ARCHIVO DE CONFIGURACIÓN PARA VER SI EXISTE.

Por tanto ahora usaremos un “Si”: * Entrada: Es mayor que *
Número: 1

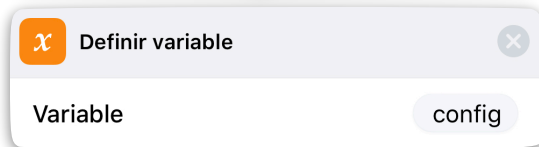
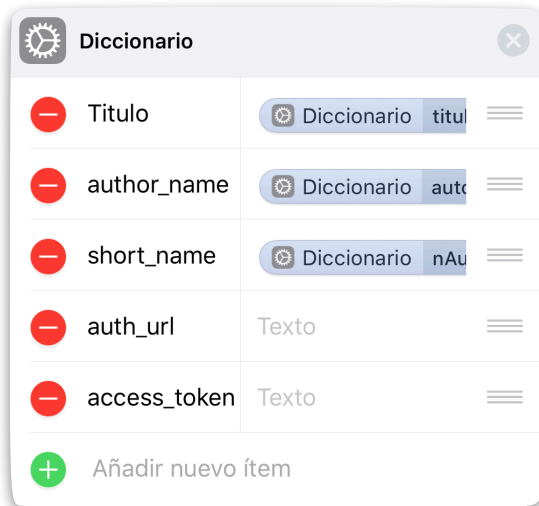
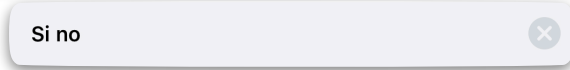
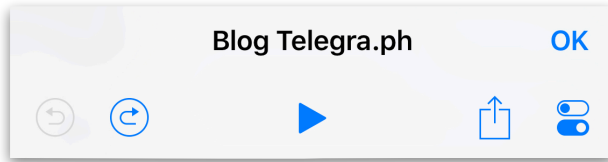
Si es la primera vez

Si existe no se debe hacer nada, ya que se guardado el archivo en la variable. En la parte de “Si no” será donde se deberá prever lo necesario para crear la estructura de archivos.

Por tanto empieza por crear un diccionario asigna los valores desde el diccionario que configuraste al principio del shortcut de la siguiente manera:

- Titulo: Diccionario(titulo)
- author_name: Diccionario(autor)
- shortname: Diccionario(nAutorCorto)
- auth_url:
- access_token:

Este nuevo diccionario se almacenara en la variable “config” mediante “Definir variable”



SE RECUPERA LA INFORMACIÓN DE LA CONFIGURACIÓN INICIAL.

A continuación se preparará la petición a la API de Telegraph usando la siguiente URL

```
https://api.telegra.ph/createAccount?  
short_name=Diccionario
```

```
(short_name)&author_name=Diccionario  
(author_name)
```

En el shortcut de ejemplo se ha puesto en un bloque de texto para que se vea más claro, pero podría estar dentro de la acción "URL" directamente.

A continuación se hará la petición mediante la acción "Obtener contenido de URL" lo que nos devolverá un texto en forma de diccionario, por lo que con "Obtener diccionario de la entrada" se puede trabajar con él.

Todas las APIs que tienen usuario y contraseña requerir de una primera petición para obtener el "token" de acceso, dicho de otra manera, la clave de autorización.

Esta es la salida de la petición:

```
{"result":  
{"short_name":"mcervantes","author_name":"Miguel de Cervantes","auth_url":"https://\/  
edit.telegra.ph/auth/  
aeE5E9GjkTLbDUhpwbHzACR5oKjpnQlv2ZBG6fZq8J","access_token":"50307c5c7b04cb9e27a3fe0209bb52b23727043bc3681259ab086f5d0199","author_url":""},  
"ok":true}
```

Entenderla así requiere de algo de experiencia con archivos JSON por lo que es recomendable usar un servicio tipo [jsonlint](#) para darle un formato mas fácil de leer con tabuladores:

```
{  
  "result": {  
    "short_name": "mcervantes",  
    "author_name": "Miguel de Cervantes",  
    "auth_url": "https://\/  
edit.telegra.ph/auth/  
aeE5E9GjkTLbDUhpwbHzACR5oKjpnQlv2ZBG6fZq8J",
```



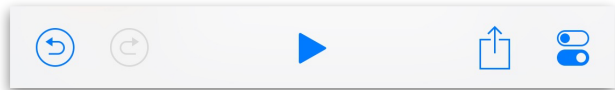
```
        "access_token":
"50307c5c7b04cb9e27a3fe0209bb52b23727043bc3681
259ab086f5d0199",
        "author_url": ""
    },
    "ok": true
}
```

Observa que el resultado es un JSON con dos claves en el primer nivel:

- result: que contiene otro diccionario con las claves:
 - short_name
 - author_name
 - auth_url
 - access_token
 - author_url
- ok: que contiene el valor booleano “true”

Por tanto para obtener el dato necesario para trabajar con la API de Telegraph es necesario acceder al segundo nivel mediante una acción “Obtener valor del diccionario” y usando el valor “result”.

Esto devolverá el segundo diccionario, por lo que hay que decirle a shortcut que lo vuelva a leer como tal mediante la acción “Obtener diccionario de la entrada” y almacenándolo en la variable “valores” ya que se necesitará para obtener más de un dato.



Texto

```
https://api.telegra.ph/createAccount?  
short_name= Diccionario short_name  
&author_name= Diccionario author_name
```

URL

URL **Texto**

Obtener contenido de URL

Avanzado

Obtener diccionario de la entrada

Obtener valor del diccionario

Obtener	Valor
Clave	result

Definir variable

Variable **valores**

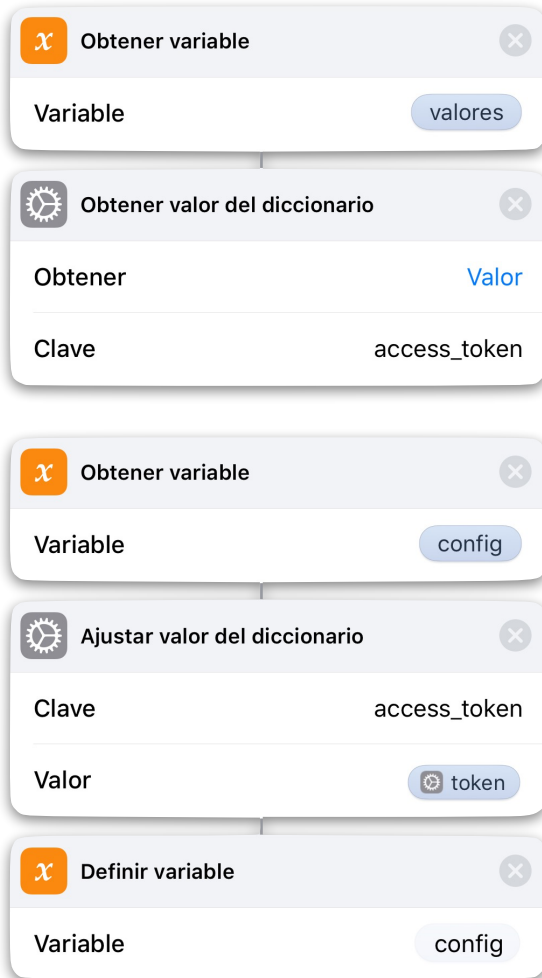
EN ESTA IMAGEN PUEDES VER CÓMO SE CONFIGURA LA PETICIÓN A LA API.

Ahora se recupera la variable en varias ocasiones y mediante la acción “Obtener valor del diccionario” se recuperarán los siguientes valores:

- auth_url
- access_token

El procedimiento en ambas el siempre el mismo:

Primero se recupera el valor correspondiente, luego la variable “config” que contenía el diccionario de configuración mediante “Obtener Variable”. A continuación se usar la acción “Ajustar valor del diccionario” para introducir el dato del diccionario de la API en nuestro diccionario de configuración. Y luego se guarda en la variable “config” para actualizar el diccionario y que contenga lo que acabas de escribir.



ESTE ES EL PROCEDIMIENTO COMPLETO PARA EXTRAER UNO DE LOS DOS VALORES QUE NECESITAMOS, EN CONCRETO EL ACCESS_TOKEN.

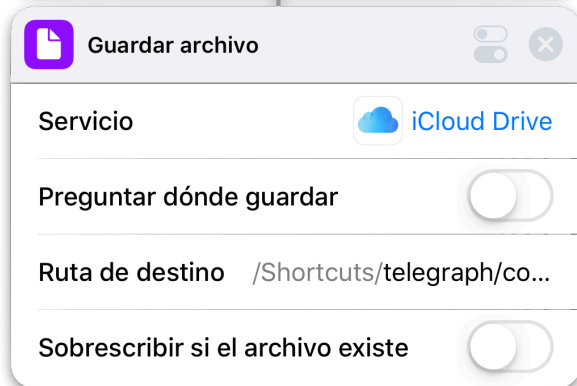
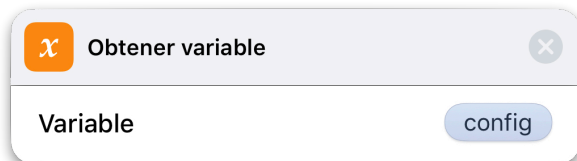
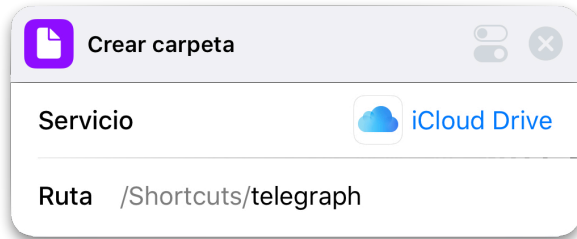
Con "auth_url" el procedimiento es exactamente el mismo.

Ahora se crea la carpeta en iCloud donde se almacenará esta configuración para recuperarla en el futuro mediante la acción “Crear carpeta”:

- Servicio: iCloud Drive
- Ruta de destino: /Shortcuts/telegraph

Recupera la variable “config” y con una acción “Guardar archivo” almacénala en iCloud:

- Servicio: iCloud Drive
- Preguntar al guardar: desactivado
- Ruta de destino: /Shortcuts/telegraph/config.json
- Sobrescribir si el archivo existe: desactivado



SE CREA LA CARPETA PARA LA CONFIGURACIÓN Y SE GUARDA EL JSON CON TODOS LOS DATOS.

Se crea la primera página En una caja de texto escribe lo siguiente que hará de encabezado de la lista de páginas que forman tu blog:

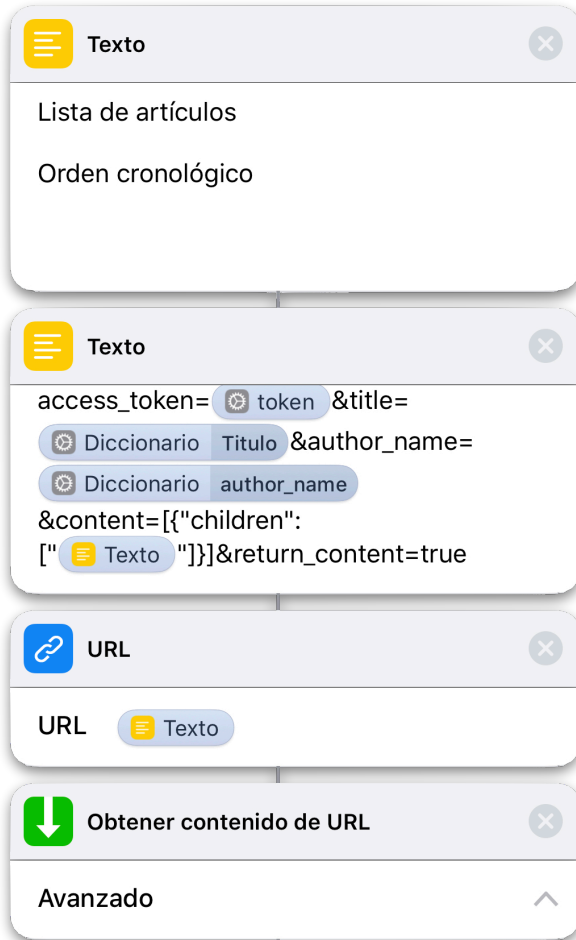
Lista de artículos

Orden cronológico

A continuación se hará una llamada a la API de Telegraph de la siguiente forma:

```
https://api.telegra.ph/createPage?  
access_token=config(access_token)&title=Diccionario (Titulo)&author_name=Diccionario  
(author_name)&content=[{"children":  
["Texto"]}]&return_content=true
```

Las variables de que deberás usar son, del diccionario “config” el `access_token` de la variable mágica “Diccionario” el título y `author_name`. Dentro del parámetro `content` el contenido es fijo [{"children":[" aquí insertamos el texto de la lista de artículos y cerramos con "]}]&return_content=true.



CON EL TEXTO DE COMIENZO DE LA PÁGINA PRINCIPAL SE HACE LA LLAMADA A LA API DE TELEGRAPH.

Se pasa a una URL y con la acción "Obtener contenido de la URL" se ejecuta.

Esta acción, como es habitual en las APIs, devuelve un JSON informando del resultado de la acción, por lo que obtendremos

el diccionario y guardaremos el valor de la clave “result” en la variable “resultadoP”. Ese nuevo diccionario tendrá una clave llamada “url”.

Obtenemos la variable “config” y añadiremos una clave llamada “index_url” a la que le asignaremos el valor “url” obtenida en el paso anterior, ya que esta va a ser la dirección de la página principal de nuestro blog en Telegraph.

De la misma manera obtendremos de “resultadoP” el valor de la clave “path” y lo almacenaremos en la variable “config” en la clave “path”. De nuevo almacenaremos todo el contenido de “config” en el archivo config.json en iCloud Drive, marcando la casilla “Sobrescribir si el archivo existe” para que guarde los cambios en el mismo archivo.

12.2.1.3. Menú de acciones

Antes de mostrar el menú de acciones se obtendrá la variable “config” para tenerla preparada para las acciones del menú.

Usa una acción “Seleccionar del menú” para añadir dos opciones al menú:

- Crear entrada
- Ver blog

Crear entrada En primer lugar usa una acción “Solicitar entrada” para que el usuario escriba el título la entrada:

- Pregunta: Título de la entrada
- Respuesta por omisión: se deja en blanco
- Tipo de entrada: Texto

Debido a las restricciones de la API para el título es necesario que los espacios se sustituyan por el símbolo más +. Así que con una acción “Reemplazar texto”:

- Buscar texto: escribe un espacio
- Reemplazar por: +

- Distinguir mayúsculas y minúsculas: no influye
- Expresión regular: desactivado

A continuación usa otra acción “Solicitud entrada” para introducir el texto que se quiere escribir:

- Pregunta: Texto del artículo
- Respuesta por omisión: se deja en blanco
- Tipo de entrada: Texto

De este texto se recuperan 35 palabras que usaremos a modo de resumen para la página principal del blog. Esto se hará con una acción “Coincidir texto”:

- Patrón: `^((?:++){35}+).*`
- Distinguir mayúsculas y minúsculas: no tiene efecto sobre la acción.

El resultado de esta acción la guardaremos en una variable “resumen” para usarla más adelante.

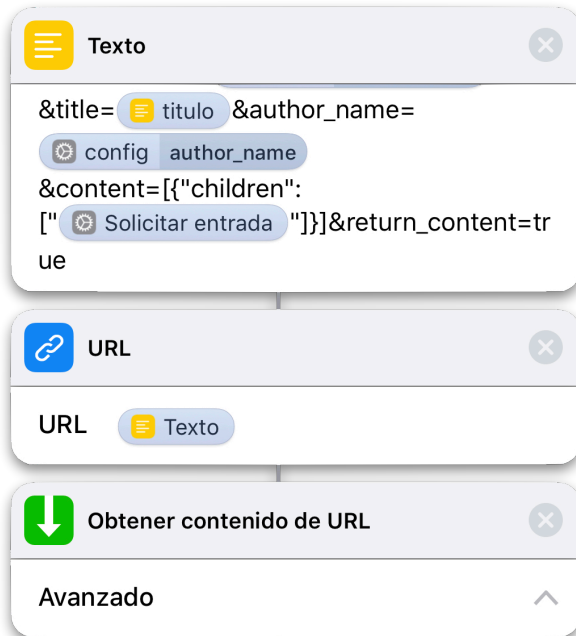
Ahora se prepara la llamada a la API de telegraph para enviar el título y el texto y que se cree la nueva página:

```
https://api.telegra.ph/createPage?  
access_token=config  
(access_token)&title=titulo&author_name=config  
(author_name)&content=[{"children":["Solicitar  
entrada"]}]]&return_content=true
```

Los parámetros de esta llamada son:

- `access_token`: Recuperación de la variable `config`, mediante la clave `access_token`.
- `title`: Usar las variables mágicas para recuperar el texto que se introdujo como título tras reemplazar los espacios por el signo más.
- `author_name`: se obtiene de la variable `config`, mediante la clave `author_name`.

- content:[{"children":[" usa la variable mágica para obtener el texto del artículo, por lo que aparecerá "Solicitar entrada" y ciérralo con]}]&return_content=true



LLAMADA A LA API DE TELEGRAPH PARA CREAR UNA PÁGINA CON UNA ENTRADA NUEVA.

Esa llamada a la API de Telegraph devuelve un JSON, en cuya clave result viene la url de la web que se acaba de crear bajo la clave url. Se recupera para añadirla a la página principal de nuestro blog.

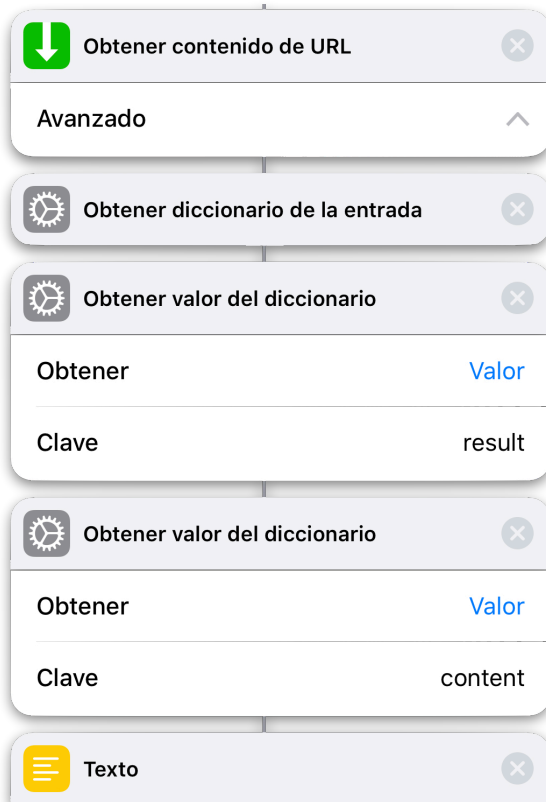
De cara a añadir la entrada en la página principal del blog se debe solicitar mediante la api el contenido de esta página con esta url:

`https://api.telegra.ph/getPage/config(path)?return_content=true`

En esta url config(path) es obtener del diccionario config la clave path. De nuevo se introduce esto en una acción "URL" y usamos la acción "Obtener contenido de URL" para hacer la llamada y obtener el resultado en JSON.

Introduce la respuesta JSON en un diccionario para obtener el contenido de la clave "result" y dentro de esta el contenido de la clave "content".

De esta manera nos devolverá el texto que ya hay dentro de la página del blog principal por lo que podremos modificarlo y volverlo a guardar una vez añadida la nueva entrada.



RECUPERA LA CLAVE RESULT Y CONTENT PARA OBTENER EL CONTENIDO DE LA PÁGINA PRINCIPAL.

Preparamos un texto con este formato:

Valor del diccionario

Solicitar entrada

resumen...

-->Seguir leyendo

Valor del diccionario

- Valor del diccionario: es la variable mágica que contiene el resultado de “Obtener valor del diccionario” con la clave “content”.
- Solicitar entrada: es el título que solicitamos al comenzar el artículo antes de sustituir los espacios por el signo más (+).
- resumen: es la variable donde se guardaron las primeras 35 palabras del artículo que acabamos de escribir.
- → Seguir leyendo: es texto escrito en la entrada
- Valor del diccionario: es la variable mágica que recupera la dirección de la página que acabamos de publicar con el artículo.

Se vuelve a introducir todo este contenido en la variable “contenido” y se realiza una llamada a la API que modifique el contenido de la página por el nuevo que se ha creado mediante esta llamada:

```
https://api.telegra.ph/editPage/config (path)?  
access_token=config  
(access_token)&title=config  
(Titulo)&author_name=config  
(author_name)&content=[{"children":  
["contenido"]}]]&return_content=true
```

En este caso ya deberías saber a qué se refiere cada parte de esta url. Se guarda en una acción “URL” y con “Obtener contenido de la URL” se hace la llamada a la API.

Ver blog

Para ver tu blog con las entradas que has ido creando solo necesitas recuperar del archivo de configuración la url de la página principal por lo que en una acción URL introduces config(index_url) y se carga el contenido con una acción del tipo “Mostrar página web”.

Se puede añadir una acción “Compartir” para que al acabar de visualizar el blog se pueda compartir el blog con quien desees.

Con esto se da por finalizado el shortcut.

Sería ideal poder haber dado formato al texto con negritas, cursivas, añadir imágenes, e incluso enlaces, sobre todo en la página principal del blog donde se enlazan todos los post que se han escrito, pero por desgracia la API de Telegraph aun no permite nada que no sea texto plano.



[DESCÁRGALO AQUÍ](#)

12.2.2. Sistema de facturación para autónomos y empresas

Como colofón a este libro se ha creado un sistema completo de facturación para autónomos y pequeñas empresas que funciona completamente desde Shortcuts.

El shortcut, como puedes imaginar es muy extenso y a la vez que repetitivo, por tanto se va a explicar someramente el esquema de funcionamiento para que sea comprensible y el

enlace de descarga para que puedas obtenerlo y verlo en profundidad.

11.2.2.1. Esquema del shortcut

El shortcut comienza con un menú que divide la ejecución en partes que son autoexplicativas de lo que hacen y todas necesarias para el funcionamiento del mismo:

- Configurar la aplicación
- Añadir cliente
- Editar cliente
- Crear nueva factura
- Ver facturas anteriores

Configurar la aplicación

Primero comprueba que no se haya configurado antes, si no lo ha hecho crea la estructura de carpetas y archivos. Si ya se ha configurado carga los datos de configuración para modificar los que necesitemos corregir.

Los datos que pide son:

- Nombre y apellidos
- DNI
- Dirección
- Código postal
- Ciudad
- Porcentaje de IVA en este momento
- Porcentaje IRPF en este momento
- Numero de cuenta en la que los clientes deberán pagar las facturas que emitamos
- Nombre del banco
- Numero de tu ultima factura para continuar la numeración en el shortcut.

Todo esto se guarda en la configuración junto con la imagen de tu firma para validar la factura que deberás tener en un

archivo de imagen del carrete y el logo tuyo o de tu empresa. No es obligatorio poner logo o firma. Funionará correctamente si no los tienes.

Añadir cliente El siguiente paso lógico era añadir un cliente para poder facturarle.

En este caso se recupera el archivo de base de datos de clientes y se piden los siguientes datos:

- Referencia del cliente
- Nombre o razón social
- DNI/CIF del cliente
- Dirección
- Código postal
- Ciudad

Una vez recopilados se añadirá a la base de datos de clientes.

Editar cliente Para poder editar los clientes que ya están en la base de datos primero se carga esta base de datos y se hará la misma solicitud de datos que cuando se añade por primera vez, pero en este caso los campos estarán rellenos para que se pueda corregir sobre lo ya escrito. Esto se hace añadiendo el valor correspondiente a “Respuesta por omisión” de la acción “Solicitar entrada”.

Una vez terminada la edición guardará los nuevos datos en la base de datos de cliente.

Crear nueva factura Crear una factura es el paso más complejo del shortcut ya que, a partir de la base imponible de cada concepto se calcula el subtotal, el IVA y el IRPF en caso de que el facturado sea persona física y no empresa. Esto último de forma automática simplemente comprobando si el DNI/CIF comienza por un número en cuyo caso será DNI (persona física) o empieza por una letra (empresa y organización).

Primero se carga la lista de facturas existente en JSON, y la lista de clientes para seleccionar al que se le va a facturar.

El siguiente paso es sumar 1 al número de factura y pedir mediante una acción "Solicitar entrada" que metas la fecha de factura, dando como fecha preseleccionada la de hoy, pues es la más probable.

Luego se piden hasta 4 conceptos de factura con su base imponible. Si pasado el primer concepto no se introducen mas conceptos, el shortcut detectará que no se desean introducir más y no preguntará por los que queden.

Para hacer esto lo que se hace es que al pedirle al usuario que introduzca el concepto de facturación se cuentan el número de caracteres y si este es 0 se saltan el resto de pasos.

Una vez están todos los conceptos se suman las bases imponibles de cada uno y se calcula el IVA y el IRPF para sumar el primero y restar el segundo. El IRPF solo se calcula, como se dijo al principio si el DNI/CIF empieza por número.

Todos estos datos se guardan en la base de datos de facturación mediante un diccionario.

Se creó una página web con un diseño de factura en html que se ha introducido en cajas de texto donde se añaden los valores del diccionario en cada campo para que aparezcan tus datos, los de tu cliente y las cantidades de la factura.

Cuando se ha unido todo, la página web se renderiza y se imprime en PDF. Este PDF se muestra al usuario para que compruebe que está todo bien. Si no está todo bien se cancela para volver a introducir los datos. Si todos los datos son correctos se guardará en el disco la base de datos de facturas con la nueva factura añadida. Se sumará 1 al recuento de facturas para que la siguiente sea correlativa. Se guardará el PDF de la factura en iCloud y se dará la opción de

enviar un correo electrónico a ese cliente con la factura adjunta.

Como puedes ver se han previsto muchas casuísticas y métodos de comprobación y recomprobación para que todo funcione bien, de ahí la complejidad de explicarlo paso a paso.

Pero te animo a que [descargues el shortcut](#) y lo estudies a fondo para ir comprendiendo el funcionamiento. Te darás cuenta de que es más repetitivo que complicado.



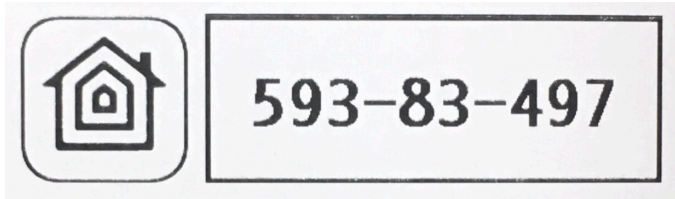
[DESCÁRGALO AQUÍ](#)

12.2.3. Copia de seguridad de tus códigos de homekit

Otra de las tecnologías que Apple tiene en marcha y que está volviendo locos a muchos es Homekit y la domótica independiente de las marcas.

En tu dispositivo no verás nada de homekit, en su lugar si que verás que hay una aplicación que se llama “Casa” ahí es donde se añaden los dispositivos compatibles para que puedan interactuar entre ellos y para que puedas encender y apagar cosas usando Siri o comandos de voz.

Todos los dispositivos compatibles con Homekit traen un código de 8 cifras que debe ser escaneado para poder unirse a la red.



ESTA ES UNA DE LAS MANERAS EN QUE SE PRESENTAN LOS
CÓDIGOS HOMEKIT

A veces serán números que la app reconocerá mediante OCR y recientemente los dispositivos más nuevos empiezan a traer un código QR codificado en cuya parte superior siempre están escritos los números para que puedas leerlos.

El sistema es bastante cómodo, normalmente éste viene en una pegatina en el dispositivo y también en el libro de instrucciones, pero tiene un problema y es que si pierdes el código jamás podrás volver a usar ese dispositivo si necesitaras volver a unirlo a la red. Ni siquiera los fabricantes tienen acceso a él, pues se crea de forma aleatoria en el momento de su fabricación.

Es por todo esto que es muy importante asegurarse de que los tienes a buen recaudo. Hay muchas soluciones: apuntarlos en una nota. Ir guardando fotos de cada uno (encontrarlas luego será otro problema). O recurrir a algunas apps que facilitar guardar este código.

En este caso te proponemos un ejercicio mediante el cual podrás generar una base de datos de tus dispositivos Homekit con sus códigos para que los tengas siempre a buen recaudo en iCloud.

Los datos con los que funcionará el shortcut se almacenarán en un archivo JSON en iCloud dentro de la carpeta Homekit que se crea a tal efecto en el primera ejecución del shortcut.

Este archivo tiene tres claves:

- Conf
- Casas
- Dispositivos

La clave **conf** tiene el valor de la ruta donde está el archivo: Homekit/conf.json para hacer referencia a él cuando sea necesario.

La clave **casas** es una lista con las casas que se añadan al shortcut, por ejemplo:

```
"casas": ["Piso", "Casa del pueblo"]
```

Y la clave **dispositivos** tiene una lista de claves que son el número de emparejamiento del dispositivo Homekit, y a cada uno de los números le corresponde un valor que es un diccionario con los datos de cada dispositivo, por lo que ese valor es, de nuevo un json.

Este es un ejemplo de como puede ser:

```
"25362536":
```

```
{  
    "modelo":"Eve Weather ",  
    "marca":"Elgato ",  
    "casa":"Piso",  
    "tipo":"Sensor",  
    "codigo":"25362536"  
}
```

Puedes ver que el código del dispositivo está duplicado, es decir, es la clave principal, pero también el valor de la clave "código" en el json secundario.

Esto se ha hecho así por el propio funcionamiento de las listas mejoradas de shortcuts. En estas listas se puede mostrar un título y un subtítulo. En este caso el título es la marca y el modelo del dispositivo: "**Elgato Eve Weather**".

Y el subtítulo es el código de emparejamiento **“25362536”**. En estas listas cuando se pulsa en una fila el valor devuelto es el subtítulo, por tanto será el número **“25362536”**. De cara a recuperar los datos de ese dispositivo desde el json para mostrar los datos en la pantalla de detalle se necesita tener bajo ese número toda la información a modo de clave.

Se podría haber creado una lista de diccionarios con los dispositivos, sin usar los códigos de los dispositivos como clave. Pero no se ha decidido hacer así porque al pertenecer a una clave, se puede hacer un pseudo borrado del dispositivo ya que si a una clave no se le asigna ningún valor esta quedará vacía: `“13214253”:{}` por lo que se le puede decir al atajo que cuando una clave esté vacía la ignore.



Recordatorio

Usando los diccionarios del shortcuts, se puede cambiar el contenido del valor de una clave, pero la clave no puede ser borrada o modificada. Es por eso que deben elegirse bien y preparar métodos para poder lidiar con claves vacías sin que suponga un problema para el funcionamiento del shortcut.

En cuanto al esquema del propio shortcut, en primer lugar contiene un menú con las siguientes opciones:

- Ver dispositivos.
- Guardar nuevo dispositivo
- Borrar dispositivo

- Añadir nueva “casa”

Añadir nueva casa simplemente pedirá el nombre con el que identifiques tu casa habitual o las casas en las que tengas los dispositivos para que sea fácil encontrarlos.

Guardar nuevo dispositivo comentará su ejecución mostrando un menú con las casas que se hayan definido en la primera acción para que el dispositivo que se agregue quede enlazado a dicha casa.

A continuación se pide la marca del dispositivo. Se mostrará una lista para la selección del tipo de dispositivo según los que hay actualmente en homekit.

Después se pide el modelo del dispositivo y su código de emparejamiento, introduciéndolo sin guiones.

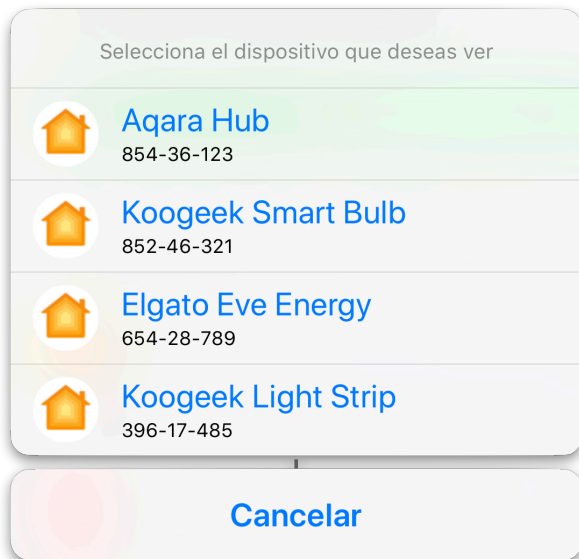
Con toda esta información el atajo genera un diccionario que será almacenado debidamente en el config.json



DICCIONARIO QUE RECOGE TODA LA INFORMACIÓN RECOPIADA A TRAVÉS DE LA INTERACCIÓN CON EL USUARIO PARA SER GUARDADA.


Borrar dispositivo empieza, de la misma manera, mediante la selección de la casa a la que está asociado el dispositivo. A continuación muestra una lista con los dispositivos disponibles en esa casa. Una vez se haya hecho clic en uno se usará su código de emparejamiento para localizarlo en el json y proceder a borrar su contenido.

Ver dispositivos lista los dispositivos una vez seleccionada la casa que queremos inspeccionar. En este caso se ha hecho uso de las listas mejoradas mediante el truco de la tarjeta de contacto para poder añadir una imagen a la izquierda, la marca y el modelo del dispositivo y el código de emparejamiento de una forma más amena.



LISTA CON LOS DISPOSITIVOS DISPONIBLES USANDO LAS LISTAS MEDIANTE TARJETAS DE CONTACTO.

Si se pulsa sobre un dispositivo concreto se puede obtener una vista aún más detallada con toda la información que se ha almacenado sobre el mismo:



Elgato Eve Energy

Ubicación	Dry
Tipo	Enchufe
Código	654-28-789

El código se ha copiado en el portapapeles

ADEMÁS, EL CÓDIGO SE COPIA AL PORTAPAPELES PARA QUE SEA FÁCIL VOLVER A INSTALAR EL DISPOSITIVO SI SE HA REINICIADO.

No se ha creado una pegatina que imite a la original porque la única tipografía que es capaz de reconocerse como propia de homekit está licenciada por Apple y no permite su uso nada más que a los fabricantes licenciados.

Descarga el shortcut, e intenta entenderlo, comenzando por lo sencillo: añadir una casa nueva. Y verás como poco a poco vas viendo que los pasos son lógicos e incluso se te pueden ocurrir mejoras ya que nunca hay una sola manera de afrontar un problema para darle solución.



[DESCÁRGALO AQUÍ](#)

12.2.4 Ejemplos

A continuación se van a listar una serie de ejemplos más cortos pero útiles donde podrás observar muchos de los conceptos explicados durante el libro. Estos ejemplos podrán servir como base para tus proyectos, o simplemente como refuerzo de todo lo visto.

1. **Detalles de Fotos:** Este shortcut muestra mediante dos métodos distintos como listar las características de una o más fotos.

Detalle de cada una de las acciones:

“Seleccionar Fotos”:

- Seleccionar varias: puedes dejar activado o desactivada esta opción. En función de si quieres que se ejecute el shortcut con una foto o con varias.

“Seleccionar del menú”:

- Ítem 1: “Cambio de tipo de variable”
- Ítem 2: “Usar bloque “Obtener Detalle de Imágenes”.

Se presentan dos opciones para que puedas ver dos técnicas distintas para llegar al mismo resultado.

Primera técnica: Cambio de tipo de variable

La primera técnica se basa en el “cambio de tipo de variable”, donde a partir de una variable, se cambia el tipo a “Contenido de Foto” y se extraen las características deseadas.

“Obtener Variable”: Se recupera la variable mágica Fotos, que ha almacenado las fotos seleccionadas.

“Repetir con Cada Uno”: Todo lo que se encuentre dentro de este bloque se repetirá con cada una de las fotos seleccionadas.

- **“Definir Variable”**: Pic. Se define la variable Pic, que almacena el ítem que en ese momento se encuentra en el “Repetir con cada uno”
- **“Texto”**: Recupera varias “propiedades” de Pic, que al clicar encima de ella se puede ver como es del tipo “foto”:

Nom:Pic (Nombre)

Fecha: Pic (Fecha de la foto)

Tamaño: Pic (Anchura) x Pic (Altura)

Screenshot:Pic (Es una captura de pantalla)

— — — — —

- **“Añadir a variable”**: info. Se recoge el texto del bloque anterior y se almacena en la variable “Info”, que es como un array de textos, es decir, es un conjunto de bloques de texto almacenados.
- Fin de **“Repetir con Cada Uno”**

Segunda técnica: Bloque “Obtener Detalle de Imágenes”

La segunda opción es más laboriosa, pero usa el bloque “Obtener detalles de Imágenes” para ir extrayendo cada una de las opciones.

- **“Obtener Variable”**: Se recupera la variable mágica Fotos, que ha almacenado las fotos seleccionadas.
- **“Repetir con Cada Uno”**: Todo lo que se encuentre dentro de este bloque se repetirá con cada una de las fotos seleccionadas.
 - **“Obtener detalles de imágenes”**: Se recupera el nombre de la foto:
 - **“Obtener variable”**: Repetir ítem. Se recupera el ítem en el que se encuentra en ese momento el bloque repetir.

Esto es necesario para que el bloque siguiente tenga como entrada la foto, y se pueda recuperar la siguiente detalle de la foto.

- “**Obtener detalles de imágenes**”: Se recupera la fecha de la foto.
- “**Obtener detalles de imágenes**”: Se recupera la Anchura.
- “**Obtener detalles de imágenes**”: Se recupera la Altura.
- “**Obtener detalles de imágenes**”: Se recupera la Si es captura de pantalla.
- “**Texto**”: Se genera un texto con toda la información recuperada usando variables mágicas.

Nom: **Nombre**

Fecha: **Fecha**

Tamaño: **Anchura** x **Altura**

Screenshot: **Escaptura**

-
- “**Añadir a variable**”: info. Se almacena cada texto en este “array” de textos para poder tener la info de todas las fotos en un mismo lugar.

Fin de “**Repetir con Cada Uno**”

“**Texto**”: Un bloque común a los dos modos de extraer la información que presenta simplemente el contenido de la variable Info, que contiene a su vez todas las características de todas las fotos seleccionadas en el primer paso.



[DESCÁRGALO AQUÍ](#)

2. A qué hora sale/pone el sol: Este shortcut se divide en dos partes.

- **Primera parte:** A partir de la posición geográfica y utilizando una API gratuita, devuelve el desfase horario UTC.
- **Segunda parte:** Se utiliza un a API que NO es necesario registro para obtener los datos de “Salida de Sol”, “Puesta de Sol” y “Horas de Luz”. Los datos obtenidos son respecto a “0 UTC”, por lo que es necesario aplicar el desfase obtenido en la primera parte, obteniendo finalmente los resultados correctos para nuestra situación geográfica.

Detalle de cada uno de los bloques:

“Obtener Ubicación Actual”: Se recupera la latitud y longitud de la posición actual.

“Texto”: Se genera una cadena de texto con el formato:

```
lat={ubicación actual}{Latitud} &  
lng={ubicación actual}{Longitud}
```

“Reemplazar Texto”: Se busca “.” y se substituye por “,”. (el formato americano utiliza el “.” como decimal)

“Diccionario”:

item 1: apikey | Valor de la Api Key

La API KEY se puede obtener mediante registro gratuito en <http://timezonedb.com>

“Obtener Variable”: Diccionario.apikey Se recupera el valor de la apiKey

“SI” Entrada=“ “

- **“Mostrar Aviso”:** Necesita de registro para poder ejecutar este shortcut
- **“URL”:** “<https://timezonedb.com/register>”
- **“Abrir Direcciones URL”**

“SI NO”

- **“Texto”**: Se crea el texto que se usará para llamar a la API:

```
http://api.timezonedb.com/v2/get-time-zone?  
key= {ApiKey del diccionario}  
&format=json&by=position& {Texto de  
“Reemplazar texto”}
```

- **“URL”**: Se llama a la API pasando los parámetros mediante “query string” usando el valor del bloque anterior **“Texto”**
- **“Obtener Contenido URL”**: Recupera lo devuelto por la llamada a la API (será un fichero JSON). Por ejemplo:

```
{“message”:“”,  
  “dstStart”:1540688400,  
  “countryName”:“Spain”,  
  “timestamp”:1549340093,  
  “gmtOffset”:3600,  
  “countryCode”:“ES”,  
  “abbreviation”:“CET”,  
  “formatted”:“2019-02-03 16:08:13”,  
  “dstEnd”:1553994000,  
  “dst”:“0”,  
  “nextAbbreviation”:“CEST”,  
  “zoneName”:“Europe/Madrid”,  
  “status”:“OK”}
```

- **“Obtener Diccionario de la Entrada”**: Recupera el JSON y lo convierte en un diccionario para poder recuperar valores.
- **“Obtener Valor del Diccionario”**:
 - **Obtener Valor**
 - **Clave**: gmtOffset

- **“Definir Variable”**: Se almacena el valor de `gmtOffset` en la variable `UtcSegundos`
- **“Calcular”**: Se divide el valor de `UtcSegundos` entre 3600 para recuperar el valor de UTC en horas.
- **“Texto”**: Se crea url para el bloque siguiente:
`https://api.sunrise-sunset.org/json? {Texto de “Reemplazar texto” que contiene la latitud y longitud con decimales en formato americano}`
- **“Obtener Contenido de URL”**: De forma automática, Shortcuts entiende que al contener una URL debe tratarla como tal y realiza la llamada a la API gratuita `sunrise-sunset`.
- **“Obtener Diccionario de la Entrada”**: Recupera el JSON y lo convierte en un diccionario para poder recuperar valores.

```
{“results”:
  {“sunset”：“5:15:29 PM”,
    “nauticaltwilightend”：“6:17:05 PM”,
    “astronomicaltwilightend”：“6:49:17 PM”,
    “civiltwilightbegin”：“6:35:19 AM”,
    “sunrise”：“7:04:09 AM”,
    “solarnoon”：“12:09:49 PM”,
    “daylength”：“10:11:20”,
    “civiltwilightend”：“5:44:19 PM”,
    “astronomicaltwilightbegin”：“5:30:21 AM”,
    “nauticaltwilightbegin”：“6:02:33 AM”
  },
  “status”：“OK”}
```

- **“Obtener Valor del Diccionario”**: El valor obtenido es un diccionario, y la variable mágica a la salida de este bloque se define como `DiccResult`.

- **Obtener Valor**

- **Clave:** results
- **“Crear Lista”:** El valor obtenido en “Results” es así mismo un diccionario, por lo que se usa la variable mágica saliente del bloque anterior (DiccResult), y se trata como un diccionario para recuperar dos valores a partir de las claves “sunrise” y “sunset”:
 - item 1: DiccResult.sunrise
 - item 2: DiccResult.sunset
 - **“Repetir con Cada Uno”**
 - **“Aplicar Formato a Fecha”:** Se aplica formato “HH:mm:ss” al ítem 1 (sunrise)
 - **“Ajustar Fecha”:** Añadir el valor de la variable mágica “HorasDesfase” en horas obtenido a la salida del bloque **“Calcular”** citado más arriba.
 - **“Añadir a Variable”:** Almacena el valor de sunrise (salida de sol) y sunset (puesta de sol) en una variable llamada Resultados.
 - Fin de **“Repetir con Cada Uno”**
- **“Obtener Variable”:** Recupera la variable Resultados.(será una lista ya que contiene dos valores.
- **“Obtener ítem de la lista”** - Primer ítem: recupera el valor de sunrise y será usado como variable mágica llamada SalidaSol.
- **“Obtener variable”** Resultados. Es una lista, y a continuación se va a extraer otro elemento de esta lista. Dicha lista debe ser el objeto entrante en el siguiente bloque.
- **“Obtener ítem de la lista”** - Último ítem: recupera el valor de sunset y será usado como variable mágica llamada PuestaSol.
- **“Mostrar Notificación”:** Muestra en pantalla la información de las variables mágicas:

- SalidaSol
- PuestaSol
- DiccResul.day_Lenght (variable mágica definida unos bloques más arriba)
- **“Terminar Si”**

En este ejemplo se ha podido comprobar como los cambios de tipo en la variable mágica “DiccResul” acortan mucho el número de bloques para obtener los valores.



[DESCÁRGALO AQUÍ](#)

3. Comprimir fotos y enviar como adjunto: Ejemplo sencillo para ver como adjuntar una foto previa eliminación de datos EXIF y redimensionar su tamaño a un mail.

Detalle de acciones:

“Seleccionar Fotos”: El usuario selecciona una o más fotos a enviar.

“Convertir Imagen”: Se convierten a JPEG y lo más importante, se eliminan los metadatos “EXIF” para aumentar la privacidad.

“Crear Archivo Comprimido”: Se comprimen las fotos ya convertidas en ZIP y se da el nombre de la fecha actual al fichero creado.

“Enviar Mensaje de Correo”: Recibe el .zip del bloque anterior y lo adjunta de forma automática al mail creado. Si no se introducen destinatarios, se preguntará al ejecutar el shortcut.



[DESCÁRGALO AQUÍ](#)

4. Traductor:

Nivel Básico: En este shortcut se utiliza el bloque **“Obtener artículo de Diffbot”** para obtener el artículo de una página web y traducirla al idioma por defecto definido en un diccionario. Si el texto es en el idioma nativo, da la opción a traducirlo a uno de los idiomas especificados en una lista. Se utiliza com oextensión y espera una página web como entrada.

DiffBot es una manera de extraer datos estructurados a partir de IA. Estas APIs tienen una parte pública que es utilizada por Shortcuts para utilizarla en el bloque llamado **“Obtener el artículo de Diffbot de la página web”**

Detalle de las acciones:

“Obtener el Artículo de diffbot de la página Web”: Recibe una página Web como entrada y recupera el artículo.

“Diccionario”:

ítem 1: nativo - Español

“Obtener Variable”: Recupera la variable mágica Artículo.

“Contar” caracteres: Recuenta los caracteres para identificar si hay o no información.

“SI” Entrada **“es mayor que”** 2

- **“Obtener Variable”:** Recupera la variable mágica Artículo.
- **“Definir Variable”:** Se salva el valor del Artículo en la variable Texto

“SI NO”

- **“Solicitar Entrada”**: Se solicita al usuario que introduzca texto a traducir, y por defecto la respuesta por omisión es lo que contenga el portapapeles.
- **“Definir Variable”**: Se salva el valor introducido en la variable Texto

“Terminar Si”

“Detectar Idioma con Microsoft”: Recibe el texto a traducir y mira en que idioma está.

“SI” Entrada es igual al valor de la clave Diccionario.nativo. (en el ejemplo “español”)

“Crear Lista”:

- Ítem 1: Inglés
- Ítem 2: Francés
- Ítem 3: Alemán
- Ítem 4: Español
- Ítem 5: Otro

“Seleccionar de la lista”: Recibe como entrada la lista del bloque anterior y el usuario debe escoger una opción.

“Definir Variable”: TraducirA

- **“SI”** Es igual a Otro.
 - **“Obtener Variable”** Texto. Recupera la variable definida anteriormente. Contiene el artículo extraído de la página Web, o lo que el usuario haya introducido.
 - **“Traducir Texto con Microsoft”**: Se presenta una lista con todos los idiomas para que el usuario seleccione su preferencia.
 - **“De”**: Detectar Idioma
 - **“A”**: Solicitar al Ejecutar
- **“SI NO”**
 - **“Obtener Variable”** Texto. Recupera la variable definida anteriormente. Contiene el artículo extraído de la página Web, o lo que el usuario haya introducido.
 - **“Traducir Texto con Microsoft”**: Se presenta una lista con todos los idiomas para que el usuario seleccione su preferencia.

- **“De”**: Detectar Idioma
- **“A”**: variable TraducirA.

“Terminar Si”

“SI NO”

“Obtener Variable” Texto

“Traducir Texto con Microsoft”: Se traduce el texto.

- **“De”**: Detectar Idioma
- **“A”**: Valor de la clave Diccionario_nativo.

“Terminar SI”

“Copiar en el portapapeles”: Copia el texto traducido.

“Leer Texto con Voz”: Lee el texto traducido.



[DESCÁRGALO AQUÍ](#)

Traductor Páginas Web: Traductor simple al igual que potente. Desde cualquier Web en inglés, se llama a este shortcut y obtendrás la página en español usando la potencia de Google, y sin API !

Detalle de acciones:

“Texto”:

`http://translate.google.com/translate?js=n&sl=auto&tl=es&u=Entrada de atajo`

“Abrir direcciones URL”: Abre la URL del texto anterior, traduciendo la página manteniendo el formato de la página.



DESCÁRGALO AQUÍ

5. **Gráfico de precios de artículos de Amazon:** Si usas Safari para moverte por Amazon, usando el menú de compartir se puede lanzar este shortcut y ver el historial de precios usando uno de estos dos shortcuts:

Keepa: Recibe una URL como entrada

”**Obtener Ítem de la lista**”: Primer ítem
”**Coincidir texto**”:

Patrón: [0-9a-zA-Z]{10}

Recupera una cadena de 10 números/letras. Los enlaces de Amazon son del tipo:

[https://www.amazon.es/xxx/dp/B07H334QWK/
ref=xx?s=...&ie=...&qid=...&sr=1-6&keywords=...](https://www.amazon.es/xxx/dp/B07H334QWK/ref=xx?s=...&ie=...&qid=...&sr=1-6&keywords=...)

”**Texto**”: URL para el bloque siguiente, donde “Coincidencias de texto” es una variable mágica que recoge el resultado del bloque “**Coincidir Texto**” anterior

<https://keepa.com/#!product/9-Coincidencias de texto>

”**Abrir Direcciones URL**”: Abre la URL y muestra los resultados.

6. **Limpiador - Validador de IBAN:** Cuando se recibe un IBAN para realizar alguna transferencia, a menudo está con “-” o “ “ molestos que dificultan realizar alguna operación bancaria. Bien, con este shortcut se puede rápidamente ver si

un IBAN es válido o no, todo previo formateo para dejarlo libre de espacios o guiones molestos.

Los detalles de cada acción a continuación:

“Diccionario”:

- ítem 1: apikey (Añadirás la ApiKey obtenida en <https://bank.codes/users/register/> de forma gratuita.

“Obtener Valor del Diccionario”: Recupera la ApiKey

“SI” Entrada es igual a “”

- **“Mostrar Aviso”:** Puedes Obtener la API KEY en la siguiente página.
- **“URL”:** <https://bank.codes/users/register/>
- **“Abrir Direcciones URL”**

“Terminar SI”

“SI” Entrada contiene Variable Mágica Especial “Entrada de Atajo”

“SI NO”

- **“Obtener Portapapeles”**

“Terminar SI”

“Definir Variable” TxT: A la salida del bloque SI/Si NO anterior se obtiene el valor de la entrada al shortcut (si proviene de iniciar el shortcut en forma de extensión, o lo que tengas en el portapapeles)

“Solicitar Entrada”: Se pregunta al usuario el IBAN, indicando como respuesta por omisión la variable Txt.

“Reemplazar Texto”: Objetivo, reemplazar todos los caracteres que no sean alfanuméricos, para así poder limpiar la cadena de “.”, “-“, etc.

Buscar Texto: “\W”

Reemplazar Por: “”

Expresión Regular: SI

Ejemplo del Regex usado:

<https://regex101.com/r/5PkmPQ/1>

“Copiar en el Portapapeles”: Se copia el IBAN “limpio” para poder dejarlo ya copiado en el caso que sea válido.

“Texto”: Se construye la URL para el bloque siguiente.

```
https://api.bank.codes/iban/?  
format=json&api_key=Diccionario  
(apiKey)&iban=Reemplazar texto
```

“URL”: Texto del bloque anterior

“Obtener Contenido de URL”: Se recupera un JSON parecido a este:

```
{“result”:
```

```
{“validation”:
```

```
{“iban_validity”:"Valid",  
“iban_structure”:"Valid",  
“iban_length”:"Valid",  
“account_checksum”:"Valid",  
“iban_checksum”:"Valid"},  
“data”:{  
“swift_code”:"BKBKESMMXXX",  
“bank_code”:"0138",  
“phone”:"091567500",  
“account_number”:"0106780195",  
“bban”:"01288712485960230195",  
“iban”:"ES3849568234560100836775",  
“countrycode”:"ES",  
“checksum”:"98",  
“fax”:"0913345522",  
“web”:"",  
“sepa”:"Yes",  
“bank”:"MIBANCO, S.A.”,
```

```

        "city":"MADRID",
        "iban_paper":"ES45 9999 9999 9999 9999
        9999",
        "branch_code":"9999",
        "country":"Spain",
        "currency":"EUR",
        "check_digit":"99"
    }
}
}

```

“Obtener Valor del Diccionario”: Se recupera “validation”

- Clave: validation

“Mostrar Aviso”: Se muestra la info que se recupera de las distintas claves:

Visibility:Validation.(iban_validity)

Structure :Validation.(iban_structure)

Lenght: Validation.(iban_length)

Ac_check:Validation.(account_checksum)

Iban_check:Validation.(iban_checksum)

“Coincidir Texto”: Se usa Regex una vez más para recuperar los parámetros del bloque anterior que tienen el string “Valid”.

Patrón: “Valid”

“Contar”: Ítems. Cuenta el número de “Valid”, si es 5, es que todo es correcto y el IBAN será correcto.

“SI” Es igual a 5

- **“Mostrar Aviso”:** Se muestra mensaje indicando que el IBAN es correcto y que está en el portapapeles.

“SI NO”

- **“Mostrar Aviso”:** Indica que el IBAN no es correcto.

“Terminar SI”



[DESCÁRGALO AQUÍ](#)

7. BackUp de tus shortcuts: Un sencillo shortcut para tener a salvo tu librería. Han habido ya numerosos casos de pérdidas de librerías por parte de algunos usuarios. Con este shortcut y una programación con DUE por ejemplo, se pueden realizar copiar semanales.

Detalle de acciones:

“Obtener mis Atajos”: Recupera un listado de toda tu librería de shortcuts.

“Crear Archivo Comprimido”: Creará un Zip con el nombre y formato deseados.

- Nombre del Archivo: Shortcuts Backup(Variable Especial FechaActual)
- Formato: .zip

“Guardar Archivo”: Salva el .zip en la nube.

- Servicio: iCloud Drive
- Preguntar Donde Guardar: NO
- RutaDestino: /Shortcuts/BackUps

“Mostrar Notificación”: mensaje indicando que se ha guardado el fichero.



[DESCÁRGALO AQUÍ](#)

8. Conversor de Moneda: Este shortcut no necesita de registro en la API utilizada, y realiza conversiones de monedas, permitiendo incluso tener una como favorita para ganar rapidez en su ejecución. Como aspecto a destacar: Después de recuperar los valores del diccionario de monedas se usa el bloque **“Filtrar Archivos”** para poder ordenarlas alfabéticamente.

Detalle de acciones:

“Diccionario”: Se definen variables.

- **SelValoresXdefecto:** Si se va a realizar siempre la misma conversión, por ejemplo EUR a USD, marcar **SI**. Por el contrario, si se van a realizar distintas conversiones, marcar **NO**.
- **cOrigen:** EUR (moneda origen)
- **cDestino:** USD (o la moneda que se desee)

“Obtener Valor del Diccionario”: SelValoresXdefecto.
Recupera el valor SI/NO

“SI” Entrada Es igual a NO. (es decir, quiere escoger moneda origen y destino)

- **“URL”:** [“https://free.currencyconverterapi.com/api/v6/currencies”](https://free.currencyconverterapi.com/api/v6/currencies)
- **“Obtener Contenido de URL”:** Se realiza la petición a la API y se recupera en formato JSON la info con el formato siguiente:

```
results: {  
  "EUR": {  
    "currencyName": "EURO",  
    "currencySymbol": "€",  
    "id": "EUR"  
  },
```

```

"USD":{
  "currencyName":"United States Dollar",
  "currencySymbol": "$",
  "id": "USD"
},
... (seguirían el resto de monedas)

```

- **“Obtener Valor del Diccionario”**: Obtener Todas las Claves. Se recuperan todos las claves del diccionario anterior. (“EUR”,“USD”..., en rojo para distinguirlas del ID)
- **“Filtrar Archivos”**: Se ordena la lista de claves del bloque anterior en orden alfabético.
 - **Ordenar por**: Nombre
 - **Orden**: A-Z
- **“Repetir con Cada Uno”**
 - **“Obtener Variable”**:Recupera la variable mágica Results.
 - **“Obtener Valor del Diccionario”**
 - **Obtener**: Valor
 - **Clave**: id
 - **“Obtener Variable”**: Recupera var.mágica InfoMoneda
 - **“Obtener Valor del Diccionario”** :
 - **Obtener**: Valor
 - **Clave**: currencyName
 - **“Obtener Valor del Diccionario”** :
 - **Obtener**: Valor
 - **Clave**: currencySymbol
 - **“Texto”**: Se agrupan los valores recuperando las variables mágicas anteriores con este formato.
 - Id - NombreMon - Símbolo
 - **“Añadir a Variable”**: Se adjunta el bloque saliente anterior a la variable Monedas.

“Terminar Repetición”

- **“Obtener variable”** : Se recupera la variable “Monedas”, que contiene todas las monedas disponibles a convertir.

- **“Seleccionar de la Lista”**: Se presenta la lista para que el usuario seleccione la moneda Origen.
 - **“Solicitar Entrada”**: Cantidad a convertir de la elección del bloque anterior. Por defecto se presenta la cantidad de 100.
 - **“Definir Variable”**: Cantidad
 - **“Obtener Variable”**: Se recupera de nuevo Monedas, para poder presentar de nuevo la lista de monedas a convertir.
 - **“Seleccionar de la Lista”**: Moneda destino, a que monede se desea convertir.
 - **“Dividir Texto”**: Se utiliza el separador personalizado “-“ para poder separar la moneda destino escogida por el usuario. Se obtiene algo parecido a esto:
 - **Id**: Identificador de moneda (Ej. EUR)
 - **Nombre** descriptivo de la moneda (EURO)
 - **Símbolo** (€)
 - **“Obtener Ítem de la Lista”**: Primer Ítem, se recupera el Id de la moneda destino.
 - **“Definir Variable”**: Destino.
 - **“Obtener Variable”**: Variable Mágica DescrMonOrigen, que es la moneda Origen escogida por el usuario. (Variable resultante a la salida del bloque “Seleccionar de la lista” (Selecciona moneda origen que se encuentra más arriba)
 - **“Dividir Texto”**: Se “parsea” (divide) de forma personalizada (“-“) la moneda origen escogida por el usuario para poder recuperar su id.
 - **“Obtener Ítem de la Lista”**: Primer Ítem. (El id de la moneda origen)
- “SI NO”** (Si se desea convertir según los valores por defecto)
- **“Texto”**: Se recupera la moneda origen definida en el diccionario inicial llamado “Setup”.
 - **“Definir Variable”**: Origen
 - **“Texto”**: Se recupera la moneda destino definida en el diccionario inicial llamado “Setup”.
 - **“Definir Variable”**: Destino

- **“Solicitar Entrada”**: Cantidad a convertir de “Origen” a “Destino” ?. Por defecto 100.

“Definir Variable”: Cantidad

“Terminar SI”

(los bloques siguientes son comunes sea cual sea la configuración de la variable SelValoresXdefecto del diccionario inicial)

“Texto”: Se construye la URL para el bloque siguiente. Se incluyen las variables definidas anteriormente, de Origen y Destino, ya provengan del “SI”, o del “SI NO”.

```
https://free.currencyconverterapi.com/api/v6/convert?
q=Origen_Destino, Destino_Origen&compact=ultra
```

“URL”: Se rellena con el bloque Texto anterior.

“Obtener Contenido de URL”: Se lanza la llamada a la API y se recupera el resultado. (en la documentación se indica que el resultado será en formato JSON y cumplirá este formato):

```
{
    "USD_PHP": 46.211,
    "PHP_USD": 0.02163987
}
```

“Obtener Diccionario de la Entrada”: Al ser un JSON, la conversión a diccionario es directa.

“Obtener Valor del Diccionario”: Se recupera el primer valor del JSON.

- **Obtener**: Valor
- **Clave**: Origen_Destino

“Calcular”: Se multiplica el valor del bloque anterior por la cantidad que el usuario quería convertir (por defecto 100)

“Obtener Variable Diccionario”: Se recupera la variable mágica Diccionario, que es la devuelta por la API, para poder extraer otro resultado.

- **Obtener**: Valor
- **Clave**: Destino_Origen

“**Calcular**”: Se multiplica de nuevo por el valor cantidad que había introducido el usuario.

“**Mostrar Aviso**”: Se presentan los resultados del cambio, y recuperando las variables mágicas ResOrDes y ResDesOri que provienen de las salidas de los bloques Calcular anteriores.

Cantidad Origen en Destino = ResOrDes

Cantidad Destino en Origen = ResDesOri



[DESCÁRGALO AQUÍ](#)

9. Escaneo de Virus en VirusTotal: En este apartado se incluyen 2 shortcuts, uno para ficheros, y otro para URLs. Este shortcut requiere de registro de forma gratuita en [VirusTotal.es](https://www.virustotal.com/), una de las mayores bases de datos del mundo de Virus.

[Escaneo de Ficheros \(32 MB máximo\)](#)

[Escaneo de URLs:](#) Acepta URLs como entrada y realiza un escaneo, devolviendo un report en HTML.

Este shortcut actúa como extensión, aceptando como entrada el tipo “Archivos”.

En detalle las acciones de **Escaneo de Ficheros en VirusTotal:**

“**Definir Variable**”: File

“**Diccionario**”:

- **apikey:** Contiene valor de la apikey, que se puede obtener de forma gratuita mediante registro en:
<https://www.virustotal.com/>

“URL”: <https://www.virustotal.com/vtapi/v2/file/scan>

“Obtener Contenido de URL”:

- **Método:** POST
- **Cuerpo de la solicitud:** Formulario
- **apikey:** Diccionario.apikey
- **file:** Entrada de atajo

“Obtener Diccionario de la Entrada”

Se obtiene una respuesta del tipo:

```
{  
  "sha256": " Francescxxxxxxxx",  
  "md5": "xxxxxxxxxxxxxxxx",  
  "scan_id": "dddddddddd",  
  "resource": "xxxxxxxxxxxx",  
  "response_code": 1,  
  "permalink": "https:\\\\xxxxxxxxxxxx",  
  "sha1": "xxxxxxxxxxxx",  
  "verbose_msg": "Scan Request successfully  
  queued, come back later for report"  
}
```

“Obtener valor del Diccionario”:

Obtener: Valor

Clave: sha256

Esperar: 10 segundos (se da un tiempo a la API para analizar y devolver los resultados)

“Texto”: <https://www.virustotal.com/vtapi/v2/file/report>

“URL”: Texto del bloque anterior

“Obtener Contenido de URL”:

- **Método:** POST
- **Cuerpo de la solicitud:** Formulario
- **apikey:** Diccionario.apikey

- **resource:** FicheroSHA (variable mágica a la salida del bloque “Obtener Valor del Diccionario” (sha256)

“**Obtener Diccionario de la Entrada**”

“**Definir Variable**”: Diccio

“**Obtener Valor del Diccionario**”:

- **Obtener:** Valor
- **Clave:** total

“**Definir Variable**”: Total

“**Obtener Variable**”: Diccio

“**Obtener Valor del Diccionario**”

- **Obtener:** Valor
- **Clave:** positives

“**Definir Variable**”: Positivos

“**Obtener Variable**”: Diccio

“**Obtener Valor del Diccionario**”:

- **Obtener:** Valor
- **Clave:** scans

“**Texto**”: Valor del Diccionario (variable mágica del bloque anterior)

“**Obtener Variable**”: Recupera la variable Positivos.

“**SI**” Entrada es mayor que 0

- “**Mostrar Notificación**”: Muestra mensaje de Virus Detectado. Se muestra la variable Total y Positivos.

“**SI NO**”

- “**Mostrar Notificación**”: Muestra mensaje indicando que el fichero está limpio.

“**Terminar SI**”



[DESCÁRGALO AQUÍ](#)

En detalle las acciones de **Escaneo de Web en VirusTotal:**

“Definir Variable”: Url

“Texto”: ApiKey de Virus Total

“URL”: <https://www.virustotal.com/vtapi/v2/url/scan>

“Obtener Contenido de URL”:

- **Método:** POST
- **Cuerpo de la Solicitud:** Formulario
 - **apikey:** ApiKey
 - **url:** Url

La API responderá con un formato del tipo:

```
{
  "scan_date": "2019-02-12 00:04:19",
  "resource": "https:\\xxxxxxx",
  "scan_id": "4564564564564644569",
  "response_code": 1,
  "permalink": "https:\\www.virustotal.com\\url\\7451axxxx",
  "verbose_msg": "Scan request successfully queued, come back later for the report",
  "url": "https:\\www.pppppp.com"
}
```

“Obtener Diccionario de la Entrada”

“Obtener Valor del Diccionario”:

- **Obtener:** Valor
- **Clave:** permalink

“Definir Variable”: Resource

“Esperar” 8 segundos

“Texto”: <https://www.virustotal.com/vtapi/v2/url/report>

“URL”: Texto del bloque anterior

“Obtener Contenido de URL”:

- **Método:** POST
- **Cuerpo de la Solicitud:** Formulario
- **apikey:** Apikey

- **resource:** Resource

“Obtener Diccionario de la Entrada”:

- Obtener: Todas las claves

“Repetir con Cada Uno”

- **“Obtener Variable”:** Recupera el valor de la variable mágica Antivirus, que procede del bloque saliente **“Obtener Valor del diccionario”** (clave scans)
- **“Obtener Valor del Diccionario”**
 - **Obtener:** Valor
 - **Clave:** Repetir Ítem
- **“Texto”:** Variable mágica “Valor del Diccionario”
- **“Texto”:**

```
<tr>
  <td style="width: 365.0625px;">Repetir ítem</td>
  <td style="width: 103.953125px;">Texto</td>
</tr>
```

“Añadir a Variable”: Tabla

“Terminar Repetición”

“Definir Variable”: Detalles

“Obtener Variable”: Diccionario.positivos

“SI” Entrada es mayor que 0

- **“Mostrar Notificación”:** Virus Detectado ! Se muestra el total de antivirus y los positivos.

“SI NO”

- **“Mostrar Notificación”:** Se muestra mensaje indicando que el fichero está limpio.

“Terminar SI”

“Texto”:

```
<h1 style="color: #5e9ca0;">URL: Url</h1>
```

```
<h4 style="color: #2e6c80;">Fecha: Fecha actual</h4>
```

```
<h2 style="color: #2e6c80;">&nbsp;</h2>
<table class="editorDemoTable" style="height:
58px; width: 487px;">
<thead>
<tr>
<td style="width: 365.0625px;">Antivirus</td>
<td style="width: 103.953125px;">Resultado</td>
</tr>
</thead>
<tbody>
Tabla
</tbody>
</table>
<p><strong>&nbsp;</strong></p>
<p><strong>M&aacute;s informaci&oacute;n en <a
href="https://www.virustotal.com/#/home/
upload">virustotal.com</a>&nbsp;</strong></p>
```

“Convertir HTML en RTF”

“Vista Rápida”



[DESCÁRGALO AQUÍ](#)

10. Ruta y distancia:

Ruta con Waze: Ejemplo sencillo en el que se utiliza la URL Scheme de Waze para obtener la ruta.

Tiempo estimado con Apple Maps: Calcula el tiempo relativo entre dos ciudades usando el coche como vehículo.

Ruta con Google Maps: Usando la URL-Scheme de Mapas de Google, en el ejemplo simple se solicita destino y abre la app para mostrar la ruta en el método de transporte escogido.

Véase cada uno de los ejemplos al detalle:

Ruta con Waze

“Solicitar Entrada”: Se pregunta al usuario la dirección de destino.

“Codificar con URL”: Codifica la dirección anterior para evitar problemas de caracteres de escape. De esta manera, el texto puede incluirse en una URL.

“URL”:

waze://?q=TextoCodificadoBloqueAnterior

“Abrir Direcciones URL”



[DESCÁRGALO AQUÍ](#)

Tiempo Estimado con Apple Maps

“Solicitar Entrada”: Se pregunta al usuario la dirección de destino

“Obtener Duración del viaje”

- **Desde**: Ubicación Actual
- **Modo de Transporte**: En coche

“Aplicar Formato a la Fecha”

“Definir Variable”: Duración



[DESCÁRGALO AQUÍ](#)

Ruta con Google Maps

“Solicitar Entrada”: Introduce la dirección de destino.

“Definir Variable”: Destino

“Seleccionar del menú”: Selecciona método de transporte.

Andando

Coche

Transporte Público

- **Andando**

 Texto: walking

- **Coche**

 “Texto”: driving

- **Transporte Público**

 “Texto”: transit

“Terminar Menú”

“Definir Variable”: Transp

“Obtener Ubicación Actual”

“Definir Variable”: CurLoc

“Obtener Detalles de Ubicaciones”: Se recupera la latitud y longitud.

- **Obtener**: Dest.Latitud - Dest.Longitud

“Reemplazar Texto”:

- **Buscar Texto**: “ ”

- **Reemplazar por**: “. ”

“Reemplazar Texto”:

- **Buscar Texto:** “-”
- **Reemplazar por:** “,”

“Definir Variable”: sLatLong

“Texto”:

comgooglemaps://?
addr=sLatLong&daddr=Destino&directionsmode=Transp

“Abrir Direcciones URL”



DESCÁRGALO AQUÍ

11. Publicar en redes sociales: Pequeño shortcut que lanzado desde Anchor o cualquier aplicación que genera un enlace permite la publicación en algunas redes sociales.

Detalles de las acciones:

“SI” Entrada Contiene “Entrada de Atajo”

“SI NO”

“Solicitar Entrada”: Introduce la URL del capítulo.

“Terminar SI”

“Definir Variable” URL

“Obtener Ítems del Canal RSS”

- **URL:** https://tufeed
- **Obtener** 1 ítem

“Obtener Nombre”

“Texto”: “

*He publicado un nuevo episodio **Nombre**
(var.mágica que corresponde a la salida del bloque anterior)*

Lo puedes escuchar aquí: **Enlace** (*Var. mágica que corresponde a la salida del bloque SI-TERMINAR SI*)

“**Solicitar Entrada**”: Se muestra el texto a publicar y se permite la edición.

• **Respuesta por omisión**: Texto

“**Añadir a Variable**”: Txlm

“**Seleccionar Fotos**”

“**Añadir a Variable**”: Txlm

“**Publicar en Twitter**”

“**Obtener Variable**”: Fotos

“**Publicar en Instagram**”: Publica la foto del bloque anterior.

Pie de foto: TextoOk (var mágica correspondiente a la salida del bloque “Solicitar Entrada” donde se muestra una previa el texto a publicar)

“**Obtener Variable**”: Txlm

“**Publicar en Facebook**”

“**Obtener Variable**”: Txlm

“**URL**”: (para publicar en un canal de Telegram)
tg://resolve?domain=nombre_bot

“**Abrir Direcciones URL**”



[DESCÁRGALO AQUÍ](#)

12. Generar código QR para WIFI: Un shortcut corto que permite generar un código QR con los datos de la wifi en la que estas conectado para que otro dispositivo se conecte sin tener que dar la contraseña.

“**Obtener detalle de la red**”:

- Red: Wi-Fi
- Obtener: Nombre de la red

“Solicitar entrada”

- Pregunta: ¿Cual es la contraseña de la wifi: {Detalle de la red}? (Variable mágica que recoge la salida de la acción anterior)
- Respuesta por omisión: se deja en blanco
- Tipo de entrada: Texto

“**Texto**”: se debe seguir el formato estándar para QR de redes wifi:

```
WIFI:S:{Detalles de red};T:WPA;P:{Solicitar entrada};;
```

{Detalles de red} es la variable mágica que recoge la salida de la primera acción “Obtener detalles de la red”.

{Solicitar entrada} es la variable mágica que recoge la contraseña que el usuario ha introducido en la acción “Solicitar entrada”.

“Generar código QR”:

- Corrección de error: medio

“**Vista rápida**” para que se muestra en pantalla el código QR y la otra persona pueda escanearlo directamente.

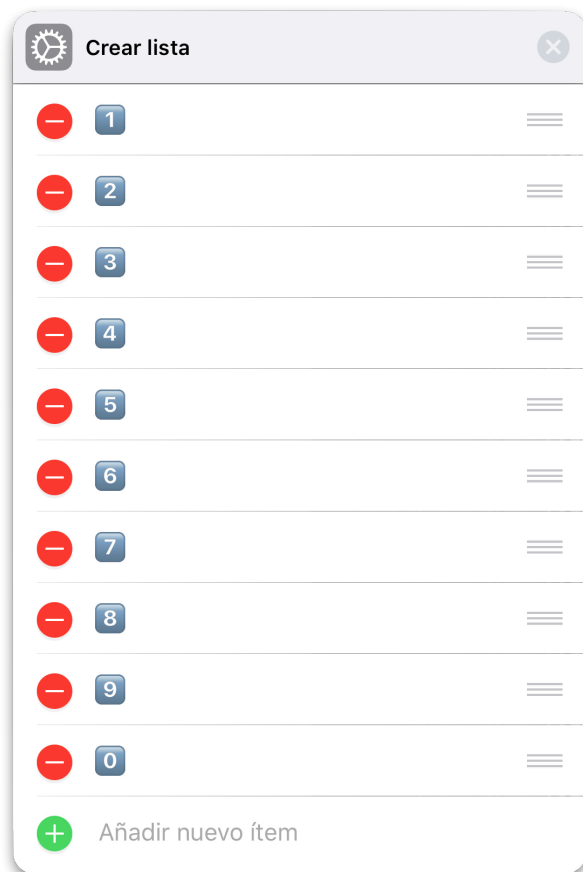


[DESCÁRGALO AQUÍ](#)

13. Saludos de buenas noches buenos días automáticos con cuenta atrás: con este shortcut se puede crear un saludo automático para enviar a un contacto o a un grupo a través de Mensajes (aunque puedes modificarlo para que sea Telegram

o WhatsApp) en el que además, en el saludo matutino se incluye una cuenta atrás a un evento que sea de interés para ambas personas o para el grupo.

“Crear lista”: Se crea una lista con los emojis de los números del 1 al 0.



LISTA CON LOS EMOJI DE LOS NÚMEROS

“Fecha”: se establece la fecha en la que ocurrirá el evento al que se quiere hacer la cuenta atrás.

- Usar: Fecha especificada
- Fecha: la que se quiera

“Obtener tiempo entre fechas”:

- Obtener tiempo desde: Ahora
- Dentro de: Días

“Dividir texto”:

- Separador: Cada carácter

“Repetir con cada uno” se va sustituyendo cada número de la cuenta atrás con su correspondiente en versión EMOJI y se especifica que si el número es 0, el elemento debe ser el 10 de la lista.

- **“Si”**

- Entrada: Es igual a
- Valor: 0
- “Obtener variable”: {Lista} (Variable mágica con el contenido de la lista de números emoji)
- “Obtener ítem de la lista”:
 - Obtener: ítem del índice
 - Índice: 10

- **“Si no”**

- “Obtener variable”: {lista} (Variable mágica con el contenido de la lista de números emoji)
- “Obtener ítem de la lista”:
 - Obtener: ítem del índice
 - Índice: Repetir ítem


- **“Teminar si”**

“Terminar repetición”

“Combinar texto”

- Separador: Personalizar
- Personalizar: (Vacío)

“Texto”:

``*`._,._,.`*`  {Texto combinado}  `*`._,._,.`*`/""`*`

{Texto combinado} es la variable mágica de la salida de la acción “Combinar texto”

Ahora discriminaremos entre si es por la mañana o por la tarde para elegir un mensaje u otro.

“Fecha”:

- Usar: Fecha actual

“Aplicar formato a fecha”:

- Formato de fecha: Ninguno
- Formato de hora: Corta

“Dividir texto”:

- Separador: Personalizar
- Personalizar: :

“Obtener ítem de la lista”:

- Obtener: Primer ítem

“Si”:

Entrada: Es menor que

Número: 14 (hace referencia a las 2 de la tarde)

“Texto”:

```
( \---/ )  
 ) .. (   
_____,---( Y )_,--._____
```

Buenos días!!!

“Añadir a variable”: Bd

“Texto”:

 Buenos  días  a  todos  !!!

“Añadir a variable”: Bd

“Texto”:

```
( \---/ )  
 ) .. (   
_____,---( Y )_,--._____
```

Buenos días!!!

“Añadir a variable”: Bd

“Texto”:

Buenos días!!!

“Añadir a variable”: Bd

“Texto”:

(__/) zZ

(>.’<)

(“)_ (“)

!!Buenos días!!

- “**Añadir a variable**”: Bd
- “**Obtener a variable**”: Bd
- “**Obtener ítem de la lista**”:
 - Obtener: ítem aleatorio.
- “**Texto**”:
{Texto}{Ítem de la lista}

{Texto} Es la variable mágica de la salida del texto que tiene la cuenta atrás

{Ítem de la lista} Es la variable mágica de la salida de la selección aleatoria del mensaje de buenos días.

“Si no”

“Texto”:

(__/) zZ

(>.’<)

(“)_ (“)

!!Buenas noches!!

- “**Añadir a variable**”: Bn
- “**Texto**”:

Me voy a la cama. Buenas noches!! ❤️❤️

- “**Añadir a variable**”: Bn
- “**Texto**”:

Buenas noches!! A dormir!! 😴😴 zZ zZ

- “**Añadir a variable**”: Bn

- **“Texto”:**

👤 Buenas 🌙 noches 🧑 a 🧑 todos 🧑 !!!

- **“Añadir a variable”:** Bn

- **“Texto”:**

Buenas noches!!

- **“Añadir a variable”:** Bn

- **“Obtener a variable”:** Bn

- **“Obtener ítem de la lista”:**

- Obtener: ítem aleatorio.

“Texto”:

{Texto}{Ítem de la lista}

{Texto} Es la variable mágica de la salida del texto que tiene la cuenta atrás

{Ítem de la lista} Es la variable mágica de la salida de la selección aleatoria del mensaje de buenos días.

“Terminar si”

“Mensajes”

Mostrar al ejecutar: desactivado

Destinatarios: el contacto al que quieras enviar el mensaje.

Mensaje: {Resultado de la acción Si} (este es la variable mágica que resulta de la ejecución de todo el shortcut).



[DESCÁRGALO AQUÍ](#)

14. Añadir emojis separando texto. Añade emojis separando las palabras del texto que uses. Este shortcut funciona como una extensión del sistema o de manera independiente introduciendo un texto. Se selecciona un texto y se le da a compartir con él y entonces se ejecuta, por lo que al comienzo se recupera el texto.

Antes de empezar se debe comprobar si el texto ha venido a través de compartir, o se va a introducir en ese momento.

“Contar”

- Recuento: Caracteres

“Si”

- Entrada: Es mayor que

- Número: 1

“Obtener variable”

- Variable: Entrada de atajo (variable mágica que recoge el texto que entra al shortcut)

“Si no”

• **“Solicitar entrada”**

- Solicitar entrada:

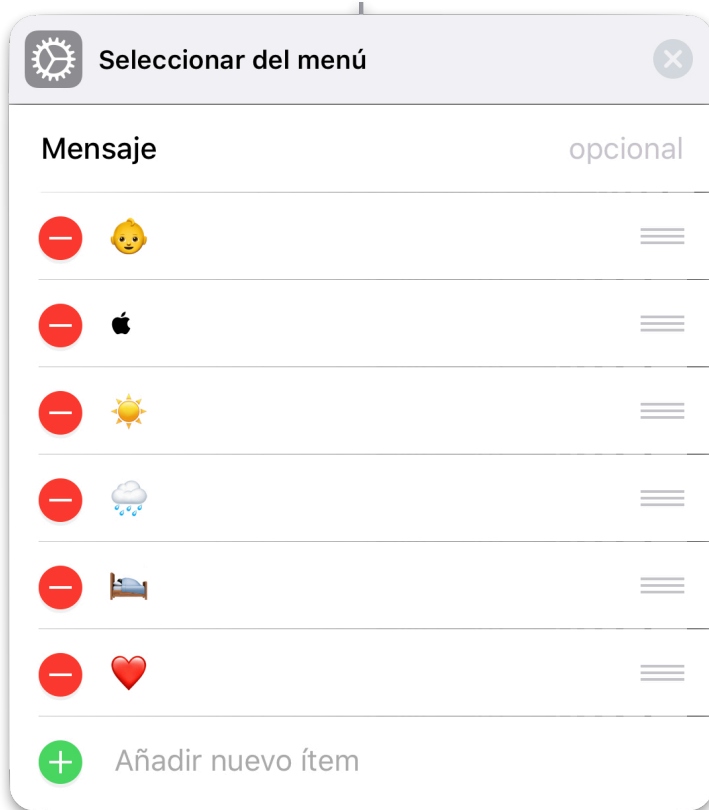
-Pregunta: Introduce el texto que vas a transformar.

“Terminar si”

“Definir variable”

- Variable: Texto

“Seleccionar del Menú” Se crea un menú con los emojis que se quieran usar para “emojizar” el texto:



LISTA DE EMOJIS QUE SE HAN ELEGIDO PARA ESTE EJEMPLO.

En cada opción del menú se inserta una acción texto que contenga el emoji se esa misma acción.

“Terminar menú”

“Obtener variable”: texto

“Reemplazar texto”:

Buscar texto: “espacio”

Reemplazar por {resultado del menu} (Variable mágica que recupera los datos del final del menú.

“Copiar en el portapapeles”

“Mostrar notificación”

Título: Añadido {Resultado del menu} al texto (Variable mágica que recoge los datos del final del menú)

Texto: Ahora puedes pegar donde quieras el texto.

El resultado de este shortcut es algo así:

Voy 🍏 a 🍏 recomendar 🍏 este 🍏 libro 🍏 a 🍏 mis 🍏 conocidos
🍏



[DESCÁRGALO AQUÍ](#)

15. Emojizador. Este shortcut es una variación del anterior en el que el emoji a usar es uno aleatorio para cada espacio en la frase de entre los introducidos en la primera acción de texto. En este caso solo funciona introduciendo la frase al ejecutar el shortcut.

“Texto”





“Dividir texto”

Separador: cada carácter

“Solicitar entrada”

Pregunta: Introduce el texto que quieres emojizar

“Dividir texto”

Separador: Espacios

“Repetir con cada uno”

“Obtener variable”: {Emojis} (Variable mágica que recupera la salida del primer dividir texto colocado justo debajo de los emojis.

- **“Obtener ítem de la lista”**

Obtener: ítem aleatorio

- **“Texto”**

{Repetir ítem} {Ítem de la lista }

“Terminar repetición”

“Combinar texto”

Separador: Personalizar

Personalizar: no se introduce nada.

El resultado de este shortcut es, por ejemplo:

Disfruto 🧐 haciendo 🦜 shortcuts 😊



[DESCÁRGALO AQUÍ](#)

16. Crear gif con las carátulas de una lista de Apple music. Selecciona una lista de reproducción al ejecutar el shortcut y obtendrás un gif animado con todas las carátulas.

“Obtener lista de reproducción”

Lista de reproducción: Solicitar al ejecutar

“Obtener imágenes de la entrada”

“Crear GIF”

Segundos por foto: 0,25

Bucle infinito: activado

Tamaño automático: activado

Con estas tres sencillas acciones el resultado puede ser algo como el siguiente gif:



RESULTADO DE LA EJECUCIÓN DEL SHORTCUT



[DESCÁRGALO AQUÍ](#)

17. Ver una app en la AppStore. Si tienes una app instalada en tu iPhone o iPad y quieres ver su ficha en la App Store para recordar que hacía o ver la lista de actualizaciones, solo tienes que pulsar su icono, y compartir con este shortcut para verla. Sencillo y útil.

“Obtener texto de la entrada”

“Obtener direcciones URL de la entrada”

“Abrir direcciones URL”

Tal y como lo ves, tres acciones encadenadas sin opciones y solucionan un problema tedioso de hacer manualmente.



[DESCÁRGALO AQUÍ](#)

18. Ver el código fuente de una web. Este es otro shortcut muy básico pero muy necesario para acceder al contenido de las webs de cara a hacer web-scraping o descubrir url de sus contenidos. Está configurado para aceptar webs desde Safari.

“**URL**”: Entrada de atajo (Variable mágica que recupera el contenido que se haya compartido con el shortcut)

“**Obtener contenido de URL**”

“**Convertir RTF en HTML**”

“**Vista Rápida**”



[DESCÁRGALO AQUÍ](#)

19. Unir imágenes en cuadrícula. A partir de una selección de fotos, crea una cuadrícula con la unión de ellas que se guarda como una sola imagen en el carrete.

“**Seleccionar fotos**”

Seleccionar varias: activado

“**Combinar imágenes**”

Modo: Cuadrícula

Espaciado: 30

“**Guardar en álbum de fotos**”



[DESCÁRGALO AQUÍ](#)

20. Com/descomprimir archivo. Este shortcut te permite seleccionar uno o varios archivos. Si es un ZIP te ofrecerá descomprimirlo donde desees. Si es uno o varios archivos y no son ZIP te ofrecerá guardar el archivo comprimido donde desees.

“Definir variable”: Archivo

“Obtener archivo”

Servicio: iCloud Drive

Mostrar selector de documentos: activado

Seleccionar varios: activado

“Repetir con cada uno”

“Añadir a variable”: Archivo

“Terminar repetición”

“Obtener ítem de la lista”: Primer ítem

“Obtener detalles de archivos”: Extensión de archivo

“Si”

Entrada: contiene

Valor: zip

- “Obtener variable”: Archivo
- “Extraer archivo comprimido”

“Si no”

“Obtener variable”: Archivo

“Crear archivo comprimido” formato ZIP.

“Terminar si”

“Guardar archivo”

Servicio: iCloud Drive

Preguntar donde guardar: Activado



[DESCÁRGALO AQUÍ](#)

21. Borrar capturas de pantalla del carrete.

“Obtener las capturas de pantalla más recientes”

Número de capturas: Solicitar al ejecutar

“Eliminar fotos”



DESCÁRGALO AQUÍ

22. Trending Topics. A través de la web trends24.in este shortcut obtiene un listado con los hashtags más utilizados de las últimas 24 horas, selecciona uno o más, y los añade al cuerpo de un mensaje personalizable que puedes publicar en Twitter.

“**Definir variable**” definimos e inicializamos la variable `no_repe`, donde guardaremos cada ítem una única vez .

“**URL**” la web de la que obtenemos los datos es <https://trends24.in/spain/>.

*nota: podemos personalizar los resultados acotándolos a un país o ciudad importante, en este caso hemos elegido España.

“**Convertir RTF en Markdown**” La web trends24.in presenta una estructura de tablas dónde aparecen los términos más utilizados en twitter agrupados en horas. Para eliminar el formato lo convertimos a Markdown.

“**Coincidir texto**” Nos quedamos con aquellas palabras que contengan el símbolo # al comienzo, los llamados hashtags. Con la expresión:

```
\[#\S+\)\(https://twitter.com/search?q=
```

Obtenemos aquellas cadenas de texto que contienen la pareja de caracteres [# , seguidas de cualquier carácter diferente al “espacio blanco”, los caracteres](y el enlace a una búsqueda en twitter.

Además habilitamos la opción que diferencia entre mayúsculas y minúsculas.

“**Reemplazar texto**” con los dos bloques siguientes limpiamos los resultados obtenidos para quedarnos únicamente con el símbolo # y la cadena que forma el hashtag. Ahora necesitamos filtrarlos uno a uno para guardar sólo un ítem de cada cadena y que no aparezcan duplicados.

“**Dividir texto**” Primero separamos el bloque de texto en líneas individuales.

“**Repetir con cada uno**” Con cada una de las líneas de texto hacemos la comprobación

- “**Obtener variable**” Recuperamos la variable no_repe
- “**Si**” En caso de que la línea que estamos procesando esté presente en la variable, seguimos sin hacer nada
- “**Si no**” En caso contrario...
- “**Obtener variable**” recuperamos el contenido de la línea de texto que estamos procesando en la variable Repetir Ítem
- “**Añadir variable**” y la añadimos al contenido de nuestra lista de hashtags únicos
- “**Terminar si**”

“**Terminar repetición**”

“**Texto**” recuperamos la lista de hashtag únicos

“**Dividir texto**” los separamos para crear una lista

“**Seleccionar de la lista**” y seleccionamos aquellos que queremos utilizar

“**Combinar texto**” volvemos a juntar únicamente los que hemos seleccionado

“**Reemplazar texto**” reemplazando el carácter \n eliminamos los saltos de línea

“**Twitter**” Finalmente abrimos una ventana de twitter para publicar un tweet que contenga aquellos hashtags que hayamos seleccionado



[DESCÁRGALO AQUÍ](#)

23. Random Users. Haciendo uso de la API de **RandomUser.me**, con este shortcut obtenemos nombres de usuario aleatorios para utilizar en registros online donde no queramos usar nuestra identidad real.

“**URL**” A partir de la web [https://randomuser.me/api/...](https://randomuser.me/api/)

“**Obtener contenido de URL**” ...hacemos una solicitud a través de su API...

“**Obtener diccionario de la entrada**” ...que nos da cómo respuesta una información aleatoria en forma de diccionario

“**Obtener valor del diccionario**”

“**Obtener diccionario de la entrada**” este diccionario contiene un sólo ítem al que podríamos llamar “persona” cuyos datos se organizan en otro diccionario

“**Definir variable**” lo guardamos como variable **xml**

“**Obtener valor del diccionario**” recuperamos el nombre a través del valor de la clave **name**

“**Definir variable**” lo guardamos como **name_dic**

“**Obtener valor del diccionario**” a su vez, la clave **name** contiene otras dos claves: **first** (nombre) y **last** (apellido), rescatamos **first**...

“**Definir variable**” ...y la guardamos en **name_1**

“**Obtener variable**” recuperamos de nuevo **name_dic**...

“**Obtener valor del diccionario**” ...para extraer el valor de la clave **last**

“**Definir variable**” ...y la guardamos como **name_2**

Ya tenemos un nombre y un apellido generados de forma aleatoria por la API del servicio randomuser.me, ahora le vamos a dar forma según queramos utilizarlo

“**Seleccionar del menú**” Presentamos un menú con dos opciones:

“**Usuario**” creamos un nombre de usuario concatenando **name_1** y **name_2** Ej: rileyhoward

“**Texto**”

“**Nombre y apellido**” separando **name_1** y **name_2** con un espacio...

“Texto”

“Cambiar mayúsculas/minúsculas” ...y convirtiendo en mayúscula la primera letra de cada una, obtenemos un nombre y apellido de la forma: Riley Howard

“Terminar menú”

“Definir variable” la solución final la guardamos en la variable **user**

“Copiar en el portapapeles” la copiamos al portapapeles para poder insertarla donde necesitemos

“Mostrar notificación” y mostramos el resultado como notificación



[DESCÁRGALO AQUÍ](#)

24. La Cafetera de Radiocable. Utilizando el feed de **iVoox** se reproduce el último programa del podcast “La Cafetera de Radiocable”, dirigido por Fernando Berlín.

“Obtener ítems del canal RSS” de la lista de programas que podemos recuperar a través del feed, nos quedamos con el último

“Repetir con cada uno”

- “Obtener variable” del ítem recuperado nos quedamos con la clave URL
- “Definir variable” la guardamos en la variable URLdelMP3

“Terminar repetición”

“URL”

“Abrir direcciones URL” abrimos la URL en el navegador para escucharlo



[DESCÁRGALO AQUÍ](#)

25. Películas hoy en TDT. Haciendo “scrapping” de la web **teletexto.com** este shortcut consigue un listado de todas las películas que se emiten en la TDT de España en abierto, y permite crear un recordatorio cuando vayan a empezar.

La web de **teletexto.com** tiene entre sus secciones una dedicada a las películas que cada día se emiten en abierto en las televisiones de España.

“**URL**” la URL de esta sección es <http://www.teletexto.com/teletexto.asp?programacion=Peliculas>

“**Obtener contenido URL**” utiliza esta URL para obtener el contenido que nos interesa

“**Convertir RTF en HTML**” convertimos el texto obtenido en formato HTML para poder rastrear el contenido utilizando las etiquetas HTML

“**Coincidir texto**” capturamos el bloque de texto que necesitamos con el patrón:

```
<table><tr><td><font>Programación de
<b>Películas</b> para hoy</font></td></tr></
table></td></tr><tr><td bgcolor=#FFFFFF>.\S </
font></td></tr></table></td></tr></table></td></
tr></table></td></tr><tr><td bgcolor=#CDCDCD
```

“**Reemplazar texto**” x 31 con esta sucesión de búsquedas y reemplazos de texto “limpiamos” el código HTML para quedarnos con la información que queremos y estructurarla en forma de diccionario.

“**Obtener valor del diccionario**” este diccionario contiene un ítem principal **cartelera**, que a su vez contiene varios elementos con la información de cada una de las películas.

“**Repetir con cada uno**” volvemos a procesar cada uno de estos elementos

- “**Obtener valor del diccionario**” recogiendo todos los valores del diccionario obtenido en la acción anterior
- “**Obtener texto de entrada**”

- **“Reemplazar texto” x 6** volvemos a limpiar a estructurar los datos en forma de lista

“Terminar repetición”

“Definir variable” a esta lista la llamamos **peliculas**

“Texto”

“Leer texto con voz”

“Obtener variable”

“Seleccionar de la lista” generamos un menú a partir de la lista

“Coincidir texto”

“Definir variable”

“Obtener variable”

“Coincidir texto” extraemos de la variable la información relacionada con la hora de emisión

“Definir variable”

“Aplicar formato a fecha” le damos formato de hora

“Texto”

“Leer texto con voz”

“Seleccionar del menú” preguntamos si queremos añadir un recordatorio

“Si”

“Añadir nuevo recordatorio”

“No”

“Terminar menú”



[DESCÁRGALO AQUÍ](#)

26. Calculadora de IVA con Soulver. Leyendo una a una cada línea de un documento la app **Soulver**, suma el importe total y calcula el IVA.

Para lanzar este shortcut necesitamos hacerlo desde el menú compartir, concretamente desde una hoja de datos de la app Soulver.

“**Dividir texto**” separamos cada una de las líneas para procesarlas individualmente

“**Repetir con cada uno**”

- “**Coincidir texto**” nos quedamos con la parte de cada línea que contiene el importe del precio
- “**Si**”
 - “**Coincidir texto**” si esta información extraída contiene el carácter €, la damos por válida
 - “**Definir variable**” y la guardamos en **valor_linea**
- “**Si no**”
- “**Terminar si**”

“**Terminar repetición**”

“**Repetir con cada uno**” ahora recuperamos la lista con los importes en euros.

- “**Calcular**” y vamos sumándolos

“**Terminar repetición**”

“**Obtener ítem de la lista**” recuperamos el total

“**Aplicar formato a número**” aplicamos formato dándole dos decimales

“**Definir variable**”

“**Calcular**” multiplicamos por 0,21 para obtener el importe del IVA

“**Aplicar formato a número**” volvemos a aplicarle formato con dos decimales

“**Definir variable**”

“**Calcular**” obtenemos el precio sin IVA por un lado

“**Texto**”

“**Mostrar notificación**” mostramos el resultado

“**Obtener variable**”

“**Copiar en el portapapeles**” y la copiamos al portapapeles para utilizarla dónde necesitemos



[DESCÁRGALO AQUÍ](#)

27. Enviar lista a Wunderlist. A partir de una lista de texto el shortcut separa cada línea y la añade como nuevo ítem de una lista de Wunderlist.

“**Solicitar entrada**” a través de un cuadro de diálogo solicitamos escribir o pegar el bloque de texto en forma de lista que vamos a exportar a Wunderlist

“**Dividir texto**” separamos cada elemento como línea individual

“**Repetir con cada uno**”

- “**Añadir tarea de Wunderlist**”

“**Terminar repetición**”



[DESCÁRGALO AQUÍ](#)

28. Abrir localización de una foto en un mapa. La mayoría de cámaras actuales y todos los móviles con cámara insertan como metadatos información sobre la posición geográfica donde se tomo la fotografía, con este sencillo shortcut vamos a extraer esa información para posicionarla en un mapa.

“**Seleccionar fotos**” seleccionamos una imagen del carrete, sólo una por ejecución

“**Obtener detalles de imágenes**” de esta imagen extraemos la información referente a la **ubicación**

“**Mostrar en Mapas**” esa dirección la podemos pasar a la aplicación **Mapas** para situar las coordenadas dónde fue tomada la imagen



[DESCÁRGALO AQUÍ](#)

29. Predicción meteorológica. Una de las consultas más recurrentes que hacemos s Siri es sobre la predicción del tiempo en nuestra ciudad. Con este shortcuts vamos a

personalizar la información que podemos obtener respecto al tiempo.

“Obtener el tiempo actual” capturamos la información meteorológica de nuestra ubicación

“Obtener detalles de ubicaciones” extraemos el nombre de la ciudad

“Texto” estructuramos esta información para que Siri nos la lea

“Leer texto con voz”



[DESCÁRGALO AQUÍ](#)

30. Acceso directo de playlist de música. Crea un acceso directo a tus playlist favoritas para reproducirlas con dos clics.

“Obtener lista de reproducción”

Lista de reproducción: selecciona la lista que quieres que se reproduzca al hacer clic en este shortcut.

“Reproducir música”



[DESCÁRGALO AQUÍ](#)

Índice de ejemplos

1	Copied - Enviar	https://www.icloud.com/shortcuts/7dc06b5f7b84485fb7d926d37e045ba7
2	Copied - Recuperación	https://www.icloud.com/shortcuts/552af82e665542bd932bcb90813e2650
3	Demo_MagicVar cambio tipos	https://www.icloud.com/shortcuts/5db4666d207d4ce7a0e065a83ee92df4
4	Demo_suma	https://www.icloud.com/shortcuts/8e5b121e84aa4e88a1be3fb982323ebc
5	Demo_MagicVar 1	https://www.icloud.com/shortcuts/e55375ce9547449b91fb231834f678b3
6	Demo_Extraccio nDetalles	https://www.icloud.com/shortcuts/8c8d6787068c4489844f1aeacb8eaa1c
7	Demo_MagicVar 2	https://www.icloud.com/shortcuts/40c2fa4da62f43b58d485f81e4e79dba
8	Demo Cambio Tipo Variable Variable	https://www.icloud.com/shortcuts/5ae0ffd6181641ca9a17cc0981edde7e
9	Variables Especiales_Entra da de Atajo	https://www.icloud.com/shortcuts/e84d69010c41495bad72442c0714065c
10	Demo var persistente	https://www.icloud.com/shortcuts/63fd07a65b1b43a9b0fc10b9215b83b3
11	Estado Series TV	https://www.icloud.com/shortcuts/97bd3b404e9d42a1a797fe57f2070996
12	Estado series TV regex	https://www.icloud.com/shortcuts/f191b2fa33524501949ed00e15148807
13	Enviar a Telegram	https://www.icloud.com/shortcuts/9b3acef64c5746088e0915aaaca234c0
14	Enviar Web a Telegram	https://www.icloud.com/shortcuts/a4533c27adca4ef2ad93eb23d127326c

15	Añadir agua a Waterminder	https://www.icloud.com/shortcuts/7ae15328ee8a46dbbda16a01da89a6fa
16	Regex_grupos	https://www.icloud.com/shortcuts/75a27bd0caea487fbb84460dc4c49e19
17	Regex_grupos 1	https://www.icloud.com/shortcuts/036c9f9f6bdd440ca0ab7230ab4fdccc
18	Últimos terremotos	https://www.icloud.com/shortcuts/5c79593f365148fc99136e9e50765371
19	Demo_unixTime- Fecha y viceversa	https://www.icloud.com/shortcuts/a7920628a6de4c5691064527d88d67e3
20	eBook Firma foto	https://www.icloud.com/shortcuts/f91a5349198f4835b57dff886cd2eb25
21	Redimensionar imagen	https://www.icloud.com/shortcuts/73fd8ba0d17e494ab3926ff5bb8abe2d
22	Reducir vídeo	https://www.icloud.com/shortcuts/c934392b848b4a598f95d1aab6f8ffab
23	Vídeo > Gif	https://www.icloud.com/shortcuts/0cd07735a1c9451ca048ef6d297e5c6c
24	Unir PDF	https://www.icloud.com/shortcuts/b08fa8d7d1f34750b17ba815c09834f2
25	Web a PDF	https://www.icloud.com/shortcuts/d97e4d28b1ea40919d0e9f55ecf0ee38
26	Comprimir y compartir	https://www.icloud.com/shortcuts/4285d20fa50a4599a076b29e719e4b19
27	Bloque Marcar (editar)	https://www.icloud.com/shortcuts/b4e90ee94e2745c2b0601e73101bb396
28	Demo_listas	https://www.icloud.com/shortcuts/f47f50f27c984b36987e97ca5737cca6

29	Demo_listas Enriquecidas 1	https://www.icloud.com/shortcuts/f39b9344045c43e5810620bd4b4c4b68
30	Demo_listas Enriquecidas 2	https://www.icloud.com/shortcuts/d5c3438444694a6b8abbd8660d128d1e
31	Demo_listas Enriquecidas 3	https://www.icloud.com/shortcuts/2ca6be7258ba474789f9ba0e88959609
32	Demo_listas Enriquecidas 4	https://www.icloud.com/shortcuts/0ac0c13df9c84fa1bb6ec6c0a929f3cf
33	Demo_listas Enriquecidas 4	https://www.icloud.com/shortcuts/7194be56fcae446abcd835b4ba6416ab
34	themoviedb info	https://www.icloud.com/shortcuts/45d0599e35a54752b81846bad12d1178
35	Codificar/ Descodificar Fotos	https://www.icloud.com/shortcuts/50d24d42ee49437c876f0b651c655d0f
36	Demo_vCard2 1	https://www.icloud.com/shortcuts/c27831f6e96d4d3b9fb68a7c8e8d0b1d
37	Demo_vCard2 1	https://www.icloud.com/shortcuts/26c57848d3824ef492a12dbb588ac0c2
38	Sub_Destinos1	https://www.icloud.com/shortcuts/eb25bb82aa024cf5ba5b780a0e7b1c6e
39	Demo_subshortcut	https://www.icloud.com/shortcuts/32a0607081564f9182f71202268acfa7
40	Diccionario sencillo	https://www.icloud.com/shortcuts/663c7f92da8040619c043f8a32ed73ac
41	Probar estructura SI	https://www.icloud.com/shortcuts/46aee64bdc9c4e5e961917f1ee96e2c1
42	demo_Optimizar Imagen Api Tinipng	https://www.icloud.com/shortcuts/c436799b96d040e08321c7ef6ce97e2d

43	Demo_CloudCon vert	https://www.icloud.com/shortcuts/ 33d6d3a29fa547b0a6874657c2c413a3
44	Demo_CloudCon vert Proteger PDF	https://www.icloud.com/shortcuts/ 5356e756bfb6491c977b8512b623a2f0
45	Demo_CloudCon vert Extensión	https://www.icloud.com/shortcuts/ 1704f5889fff4f0b8b6a60e8f4e1b126
46	Publicar en Mastodon	https://www.icloud.com/shortcuts/ 747ac5fba25948148c45bdc3c27f8309
47	OCR a Fichero y crear PDF	https://www.icloud.com/shortcuts/ 04fbe4865e8f4d839165a5b1feb998c3
48	Ejemplo 11.1.1	https://www.icloud.com/shortcuts/ 8a5528b421094050abd261e8f8a1e1f0
49	12.1.2 Ordenar Diccionario	https://www.icloud.com/shortcuts/ 558216116e6f4bc1a9f614fa0e0a5314
50	Demo_subshortc ut 3	https://www.icloud.com/shortcuts/ 60bf5d22f4d54b14b8630489427d08cc
51	Sub_Destinos 2	https://www.icloud.com/shortcuts/ c9f78a9cbc314d4e99f71c6b980994c9
52	Blog Telegra.ph	https://www.icloud.com/shortcuts/ 92620531ca384366937c8d366a3f2d0d
53	Facturas	https://www.icloud.com/shortcuts/ 9ddcfbd86a314437a86d2a74a44369e1
54	Homekit	https://www.icloud.com/shortcuts/ cb302c6494c44386b08e38a9bafd47cd
55	Detalles de Fotos	https://www.icloud.com/shortcuts/ 5be20fb99dc044f8903f14abf7decfb1
56	Demo_Puesta- Salida de Sol	https://www.icloud.com/shortcuts/ f8796a97270d492e931f2b8e466355fe

57	Seleccionar imágenes comprimir y enviar	https://www.icloud.com/shortcuts/24ac04a9101f4bddb8b7fd0246729db9
58	Traductor	https://www.icloud.com/shortcuts/7069cf9e87b248d7a943f6c931b9721e
59	Bateria2x100 traductor pagina web	https://www.icloud.com/shortcuts/5563d7135ea14d3eb24755031eb4a01e
60	Demo_IBAN VALIDATOR-CLEANER	https://www.icloud.com/shortcuts/14347351e19a42a384d628f6b34a826a
61	Shortcuts_Backup	https://www.icloud.com/shortcuts/76fd14549c49470392b2a53443250d89
62	Conversor Monedas con ARS! 1	https://www.icloud.com/shortcuts/4654e166f1604efbb1f40bdeaba3dde1
63	Demo_Virus Total v2 File	https://www.icloud.com/shortcuts/2e2eb5030e5d496c98f7476d8ae42e2d
64	Demo_Virus Total - URL	https://www.icloud.com/shortcuts/8bfdfcea8b064597a911ea142b6a9e93
65	Ruta en Waze	https://www.icloud.com/shortcuts/212488e6441e4429b70b22e631ac95fb
66	Demo_Tiempo llegada Apple maps	https://www.icloud.com/shortcuts/d920fdf0ba374335b6bfb757716f5f89
67	Demo_Ruta Con GoogleMaps(URL SCHEME)	https://www.icloud.com/shortcuts/5d980aec0e39423ead024a4b4d0be82a
68	Publicar Redes Sociales	https://www.icloud.com/shortcuts/49fa1518da1a45ca9ff7d79f96bad0bc

69	QR WiFi	https://www.icloud.com/shortcuts/8b85c7726b5d4a1a85c5f49003ea1220
70	bDbN	https://www.icloud.com/shortcuts/6183fec959344bedbbb46208abbea865
71	Añadir emojis	https://www.icloud.com/shortcuts/e00cd66f9fc74752b38c7ae96dcb286c
72	Emojizador	https://www.icloud.com/shortcuts/ca544a048ce0458a920f19450dd674f1
73	Playlist GIF	https://www.icloud.com/shortcuts/8cf2b24ac95248fab762ae82bebd1839
74	Apps-tore	https://www.icloud.com/shortcuts/3eccb9dd18d644c19bd97f2737d0fdce
75	Código fuente	https://www.icloud.com/shortcuts/b0ba6a26d5b449c9b637dfd8228ba95d
76	Cuadrícula de fotos	https://www.icloud.com/shortcuts/da0a4293f95f450794bca2fe602da28e
77	Com/ descomprimir archivo	https://www.icloud.com/shortcuts/cf2b63a07d5a4770a32c4c11941b97aa
78	Eliminar capturas de pantalla	https://www.icloud.com/shortcuts/f7c0aadb98b3476db3e7d9057783af24
79	Trending Topics	https://www.icloud.com/shortcuts/9269b2cf8fa140968282941e22fa2076
80	Random Users 2.0	https://www.icloud.com/shortcuts/c74413437e7e45608e3ed880a636a26b
81	La Resistencia	https://www.icloud.com/shortcuts/ed59472b55cc496ea4e81ab8150f4166
82	Teletexto	https://www.icloud.com/shortcuts/a37f16cc511444e9a1183ca5d70a62da

83	IVA Soulver	https://www.icloud.com/shortcuts/fd0dab52540d46aba986182f46a4791f
84	Lista Wunderlist	https://www.icloud.com/shortcuts/de5e7f3c9408494694a2d056a718f4e5
85	Localizar foto en un mapa	https://www.icloud.com/shortcuts/a5805e3c833f4eaa9e0a5c7d5989d399
86	Predicción metereológica	https://www.icloud.com/shortcuts/02b4e305142f48f3987935ff7a2498d9
87	Reproducir lista	https://www.icloud.com/shortcuts/808f55505343481cba0d1c01a6dfb9b2