



EBook Gratis

APRENDIZAJE

Xcode

Free unaffiliated eBook created from
Stack Overflow contributors.

#xcode

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con Xcode.....	2
Observaciones.....	2
Versiones.....	2
Examples.....	3
Empezar.....	3
Usa múltiples versiones de Xcode.....	4
Cambiando el esquema de color.....	6
Pro Consejo.....	8
Capítulo 2: Características de Xcode 8.....	10
Observaciones.....	10
Examples.....	10
Literales de color e imagen.....	10
Capítulo 3: Certificados, perfiles de aprovisionamiento y firma de código.....	11
Examples.....	11
Elija el enfoque de firma de código correcto.....	11
Usando la característica de firma de código de Xcode.....	11
Xcode 7 y más bajo.....	11
Xcode 8 y más.....	12
A mano.....	12
Usando el partido de Fastlane.....	12
Capítulo 4: Consejos de Xcode.....	13
Examples.....	13
Reutilizar fragmentos de código en Xcode.....	13
Instalar complementos en Xcode 7.....	14
Instalación.....	14
Recomendaciones.....	14
Uso.....	15
Oculta extraños registros Xcode 8 no deseados y extra.....	16

Capítulo 5: Creando controles personalizados en Interface Builder con @IBDesignable	18
Observaciones.....	18
Examples.....	18
Una vista redondeada en vivo.....	18
Capítulo 6: Depuración	21
Examples.....	21
Puntos de interrupción.....	21
Depuración inalámbrica en Xcode-9.....	24
Capítulo 7: Desarrollo multiplataforma	26
Examples.....	26
TargetConditionals.....	26
Capítulo 8: Herramientas de línea de comandos	28
Examples.....	28
Pruebas de carrera.....	28
Lista de objetivos disponibles, esquemas y configuraciones de compilación.....	28
Compilar y firmar esquema.....	29
Acceda a cualquier herramienta de línea de comandos en el paquete de aplicaciones Xcode (x.....	30
Cambio de herramientas de línea de comandos con xcode-select.....	30
Capítulo 9: Parques infantiles	32
Examples.....	32
Primeros pasos con Playground.....	32
Último valor, valor histórico y gráfico.....	33
Adición de imágenes, datos estáticos, sonidos, etc. a un patio de juegos.....	34
Capítulo 10: Personalizando Xcode IDE	35
Introducción.....	35
Examples.....	35
Abrir Terminal en la carpeta del proyecto Xcode actual.....	35
Borrar datos derivados con teclas de acceso rápido.....	38
Capítulo 11: Proyectos y espacios de trabajo	40
Examples.....	40
Resumen de proyectos.....	40
Crear un proyecto.....	40

Trabajando con proyectos.....	40
Construye, ejecuta, perfila, prueba y analiza tu proyecto.....	42
Ajuste el espacio de trabajo a sus necesidades y navegue libremente por él.....	43
Creditos	46

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [xcode](#)

It is an unofficial and free Xcode ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Xcode.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con Xcode

Observaciones



Xcode es un [entorno de desarrollo integrado](#) para macOS que admite el desarrollo de aplicaciones nativas para macOS, iOS, watchOS y tvOS. Xcode es el sucesor de [Project Builder](#) y PBX de NeXT. (De hecho, los archivos de manifiesto del proyecto de Xcode aún se nombran con la extensión `.pbxproj`).

Comunicados de Xcode incluyen versiones estables del [sonido metálico](#) C / C ++ / Obj-C compilador, el [Swift](#) compilador, la [LLDB](#) depurador, y simuladores IOS / watchos / TVOS. Xcode también incluye [Interface Builder](#), así como herramientas para ver y editar modelos y escenas en 3D, recursos de imagen y más.

Versiones

Versión	Fecha de lanzamiento
1.0	2003-09-28
2.0	2005-04-04
3.0	2007-10-26
4.0	2011-03-14
5.0	2013-09-18
6.0	2014-09-17
7.0	2015-09-16
7.1.1	2015-11-09
7.2	2015-12-08
7.2.1	2016-02-03
7.3	2016-03-21
7.3.1	2016-05-03
8.0	2016-09-13

Versión	Fecha de lanzamiento
8.1	2016-10-27
8.2	2016-12-12
8.2.1	2016-12-19
8.3	2017-03-27
8.3.1	2017-04-06
8.3.2	2017-04-18
8.3.3	2017-06-05

Examples

Empezar

- [Descarga Xcode](#) desde la Mac App Store.
- Haga clic para crear un nuevo proyecto o área de juegos:



Welcome to Xcode

Version 7.3 (7D175)



Get started with a playground

Explore new ideas quickly and easily.



Create a new Xcode project

Start building a new iPhone, iPad or Mac OS X app.



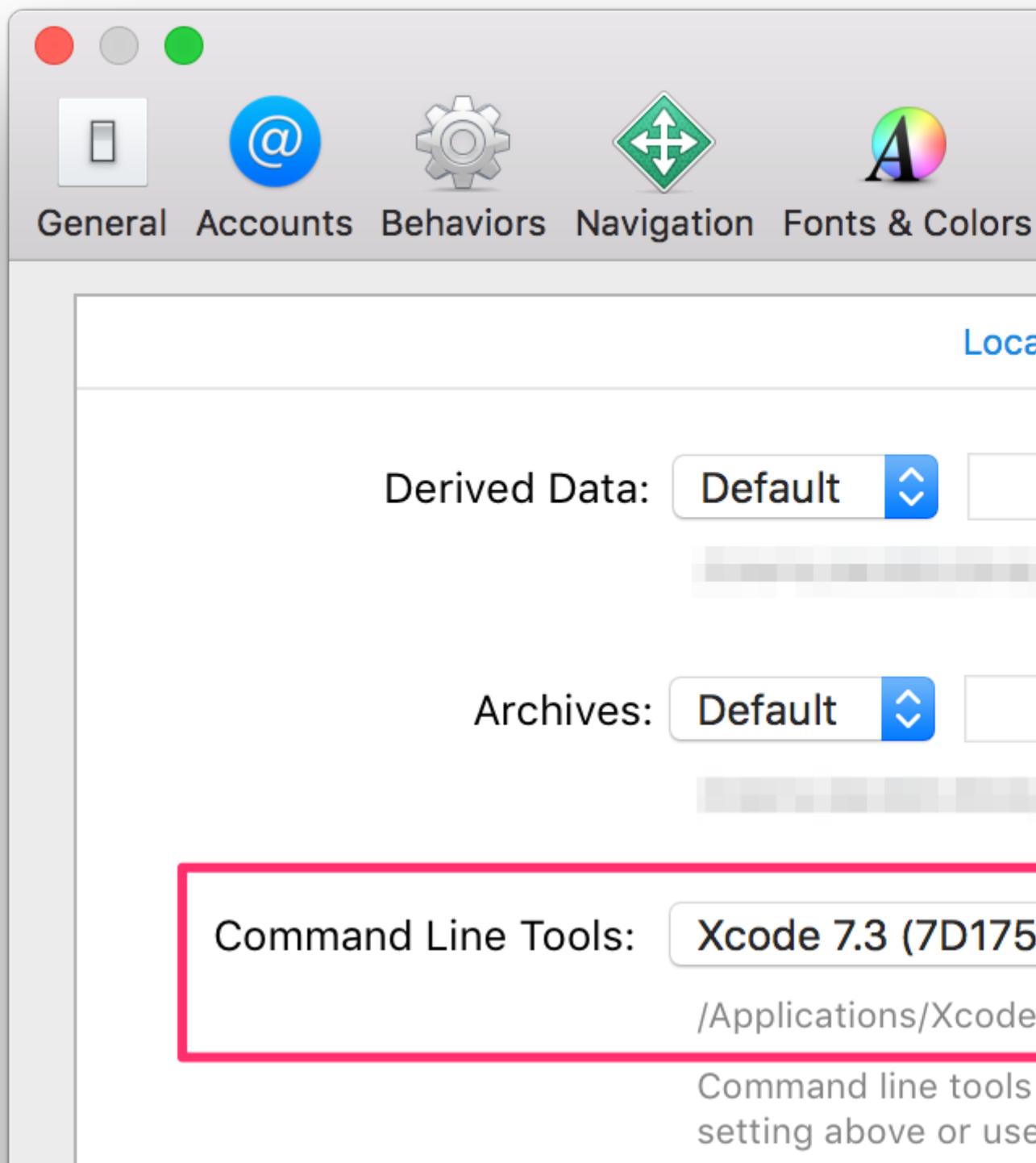
Check out an existing project

Start working on something from an SCM repository.

una versión existente en su máquina. También puede instalar Xcode desde una [descarga directa](#) para tener más control sobre qué versiones tiene.

Cada copia de Xcode incluye herramientas de línea de comandos (`clang` , `xcodebuild` , etc.). Puede elegir cuáles son invocados por los comandos en `/usr/bin` .

En las preferencias de Xcode, en la pestaña Ubicaciones, elija una versión de Xcode:



O puede administrar versiones desde la línea de comando usando `xcode-select` :

```

# Print the currently selected version
$ xcode-select --print-path
/Applications/Xcode.app/Contents/Developer

$ clang --version
Apple LLVM version 7.3.0 (clang-703.0.29)
Target: x86_64-apple-darwin15.4.0
Thread model: posix
InstalledDir:
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin

# Find all installed versions using Spotlight
$ mdfind 'kMDItemCFBundleIdentifier = "com.apple.dt.Xcode"'
/Applications/Xcode.app
/Applications/Xcode72.app

# Check their version numbers
$ mdfind 'kMDItemCFBundleIdentifier = "com.apple.dt.Xcode"' | xargs mdls -name kMDItemVersion
kMDItemVersion = "7.3"
kMDItemVersion = "7.2.1"

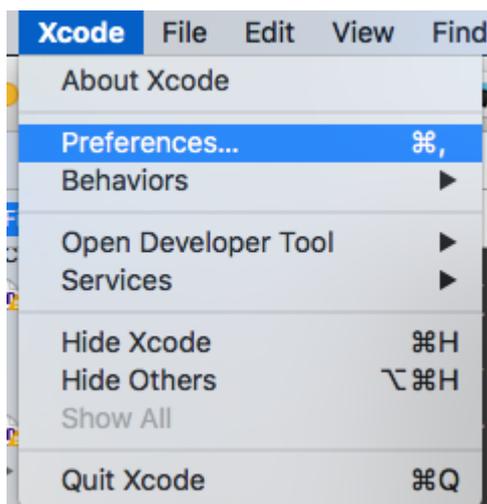
# Switch to a different version
$ sudo xcode-select --switch /Applications/Xcode72.app

$ clang --version
Apple LLVM version 7.0.2 (clang-700.1.81)
Target: x86_64-apple-darwin15.4.0
Thread model: posix

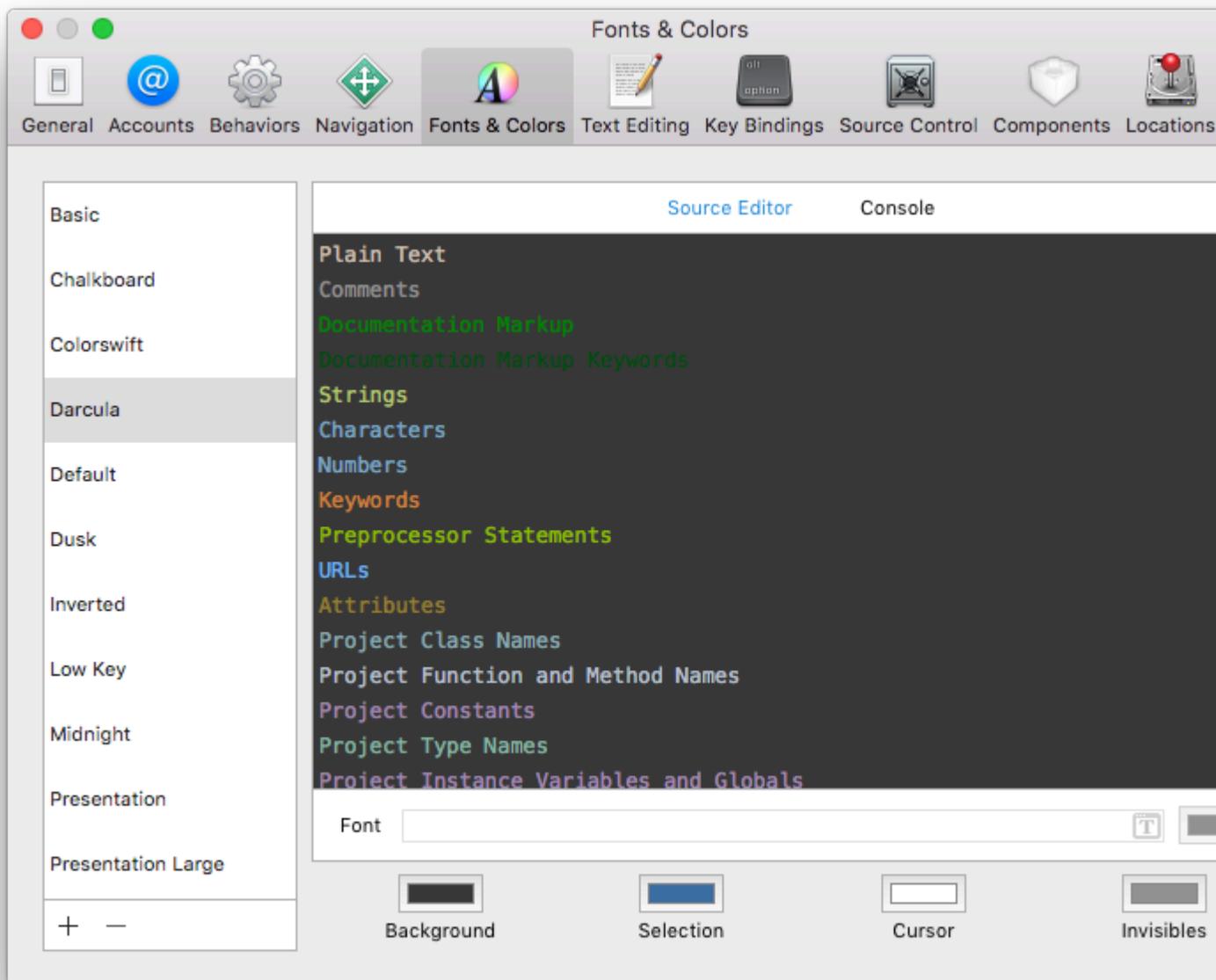
```

Cambiando el esquema de color

A muchos desarrolladores les gusta personalizar la fuente, el texto y el color de fondo de sus IDE. Puedes hacer esto en Xcode abriendo el panel de preferencias de la aplicación, ya sea yendo a XCODE-> Preferencias, o presionando "

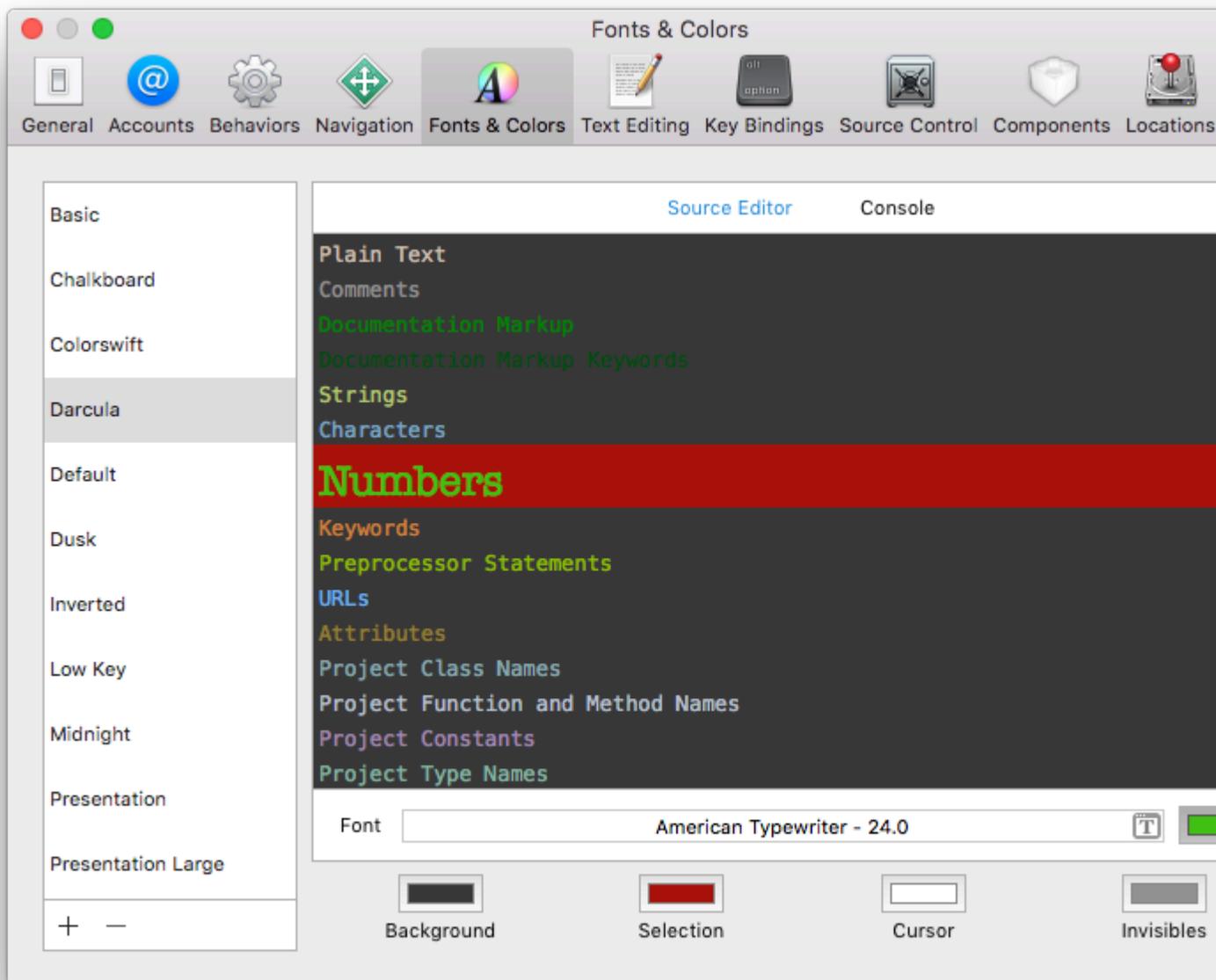


Con el panel de preferencias abierto, puede hacer clic en la pestaña 'Fuentes y colores'.



Desde aquí puede cambiar la fuente y los colores de fondo y fuente de la consola. Hay muchos esquemas de fuente y color pre-hechos provistos con Xcode. Los elige de la lista de la izquierda (Básico, Pizarra, etc.). Puede encontrar y descargar más en línea (como [aquí](#), por ejemplo).

Para personalizar aún más cualquier tema, puede personalizar cualquiera de los tipos enumerados en el panel derecho (Texto sin formato, Comentarios, Marcas de documentación, etc.). Por ejemplo, digamos que realmente quiero que mis 'Números' aparezcan en mi código. Así que cambio la fuente a 'American Typewriter' a 24 px, el color a un color verdoso, y configuro la línea resaltada en rojo:



Ahora en mi edición de texto, realmente puedo ver mis números:

```
// Dispose of any resources that can be recreated.  
let myNumber = 5;
```

¡Ahora puede personalizar la apariencia del 'Editor de código fuente' y la 'Consola' para deleite de su corazón!

Pro Consejo

A muchos desarrolladores les gusta el tema de sus IDS oscuros (texto claro, fondo oscuro). En Xcode, solo puedes hacer esto para el 'Editor de fuente' y la 'Consola'. Sin embargo, las secciones de Navegación (lado izquierdo), Depuración (parte inferior) y Utilidad (extremo

derecho) no son personalizables. Hay dos formas de trabajo para esto. Primero (un poco complicado, es dejar el tema IDE claro (fondo claro, texto oscuro) y luego invertir los colores de la pantalla a la vez. Esto hará que todo sea oscuro, pero los colores en el simulador y en el resto del sistema ahora son confusos. La segunda solución es ocultar las áreas de Navegación, Depuración y Utilidad cuando no esté en uso. Puede alternar estas áreas rápidamente usando los siguientes comandos:

Navegador: 0

Área de depuración: Y

Utilidad: 0

Lea Empezando con Xcode en línea: <https://riptutorial.com/es/xcode/topic/294/empezando-con-xcode>

Capítulo 2: Características de Xcode 8

Observaciones

Esto solo funciona con proyectos usando Swift 3+

Examples

Literales de color e imagen.

Xcode 8 reconocerá automáticamente cualquier imagen que tenga en un Catálogo de Activos y las ofrecerá como una sugerencia dentro de un inicializador UIImage.

Así que básicamente podría declarar una nueva variable y luego agregar un nombre de activo que haya agregado a su catálogo de activos. Por ejemplo, `let img = dog . img` ahora contiene la imagen de `dog` en el catálogo de activos.

Bajo el capó está creando un código que se ve así: `#imageLiteral(resourceName: "dog.png")`. Pero en línea en el editor de origen, solo verá el nombre de archivo de la imagen.

Así que podrías hacer esto ahora `imageView.image = img`.

Tenga en cuenta que debe hacer clic en la sugerencia de sentido instantáneo para que pueda ver una miniatura de la imagen en el código y luego el nombre de la imagen.

Lea Características de Xcode 8 en línea:

<https://riptutorial.com/es/xcode/topic/7155/caracteristicas-de-xcode-8>

Capítulo 3: Certificados, perfiles de aprovisionamiento y firma de código

Examples

Elija el enfoque de firma de código correcto

Si está comenzando un nuevo proyecto, es importante pensar en cómo desea manejar la firma de código.

Si eres nuevo en la firma de código, revisa la [sesión de WWDC](#) que describe los aspectos básicos de la firma de código en Xcode.

Para firmar correctamente el código de su aplicación, debe tener los siguientes recursos en su máquina local:

- La clave privada (archivo `.p12`)
- El certificado (archivo `.cer`), que coincide con la clave privada.
- El perfil de aprovisionamiento (archivo `.mobileprovision`), que coincide con el certificado y la clave privada instalada localmente

En el Portal de desarrolladores de Apple también se requiere tener una ID de aplicación válida asociada a su perfil de aprovisionamiento.

Usando la característica de firma de código de Xcode

Ocasionalmente, la configuración `Automatic` ya que el perfil de aprovisionamiento no funciona de manera confiable, ya que solo seleccionará el perfil de aprovisionamiento actualizado más reciente, sin importar si el certificado está instalado.

Por eso se recomienda especificar un perfil de aprovisionamiento específico de alguna manera:

Xcode 7 y más bajo

Debe evitar hacer clic en el botón `Fix Issue` (hay un [complemento de Xcode](#) que deshabilita el botón), ya que a veces revoca los certificados existentes y con ellos los perfiles de aprovisionamiento.

Lamentablemente, no puede especificar el nombre del perfil de aprovisionamiento en Xcode 7. En su lugar, puede especificar el UUID del perfil, que cambia cada vez que el perfil se vuelve a generar (por ejemplo, cuando agrega un nuevo dispositivo).

Para [solucionar](#) este problema, consulte [XcodeProject.md](#) sobre cómo pasar un perfil de aprovisionamiento a Xcode al crear su aplicación.

Xcode 8 y más

Apple mejoró mucho la firma de código con el lanzamiento de Xcode 8, lo siguiente ha cambiado:

- No más el botón `Fix Issue`, en su lugar, todos los procesos de firma de código se ejecutan en segundo plano y muestran el registro directamente en Xcode
- Ahora puede especificar el perfil de aprovisionamiento por nombre, en lugar del UUID ([consulte XcodeProject.md](#) para obtener más información)
- Mensajes de error mejorados cuando algo sale mal. Si se encuentra con errores de firma de código, siempre debe intentar construir y firmar con Xcode para obtener información de error más detallada. (Echa un vistazo a la [solución](#) de [problemas.md](#) para obtener más información)

A mano

Siempre puede crear y administrar manualmente sus certificados y perfiles de aprovisionamiento utilizando el Portal de desarrolladores de Apple. Asegúrese de almacenar la clave privada (`.p12`) de sus certificados en un lugar seguro, ya que no se pueden restaurar si los pierde.

Siempre puede descargar el certificado (`.cer`) y el perfil de aprovisionamiento (`.mobileprovision`) desde el Portal de desarrolladores de Apple.

Si revoca su certificado o caduca, todos los perfiles de aprovisionamiento asociados no serán válidos.

Usando el partido de Fastlane

El concepto de [coincidencia](#) se describe en la [guía de](#) firma de códigos y es el método de firma de códigos recomendado si usa [fastlane](#)

Con [Match](#) usted almacena sus claves privadas y certificados en un repositorio de git para sincronizarlas en las máquinas. Esto facilita la incorporación de nuevos miembros del equipo y la configuración de nuevas máquinas Mac. Este enfoque [es seguro](#) y utiliza la tecnología que ya utiliza.

Para comenzar con [Match es](#) necesario que revoque sus certificados existentes.

[Lea Certificados, perfiles de aprovisionamiento y firma de código en línea:](#)

<https://riptutorial.com/es/xcode/topic/3711/certificados--perfiles-de-aprovisionamiento-y-firma-de-codigo>

Capítulo 4: Consejos de Xcode

Examples

Reutilizar fragmentos de código en Xcode

Puede guardar sus fragmentos de código para usarlos más tarde simplemente arrastrando y soltando. Por ejemplo: si tiene una declaración NSLog que se usa para tantos lugares en algún otro lugar del proyecto, puede guardar las declaraciones NSLog en la biblioteca de fragmentos de código.



Xcode

File

Edit

View

Find

Navigate



Sample >



iPhone 6s Plus



S

▼ Sample

▼ Sample

h AppDelegate.h

m AppDelegate.m

h ViewController.h

m **ViewController.m**

Main.storyboard

Assets.xcassets

LaunchScreen.storyboard

Info.plist

▶ Supporting Files

▶ Products

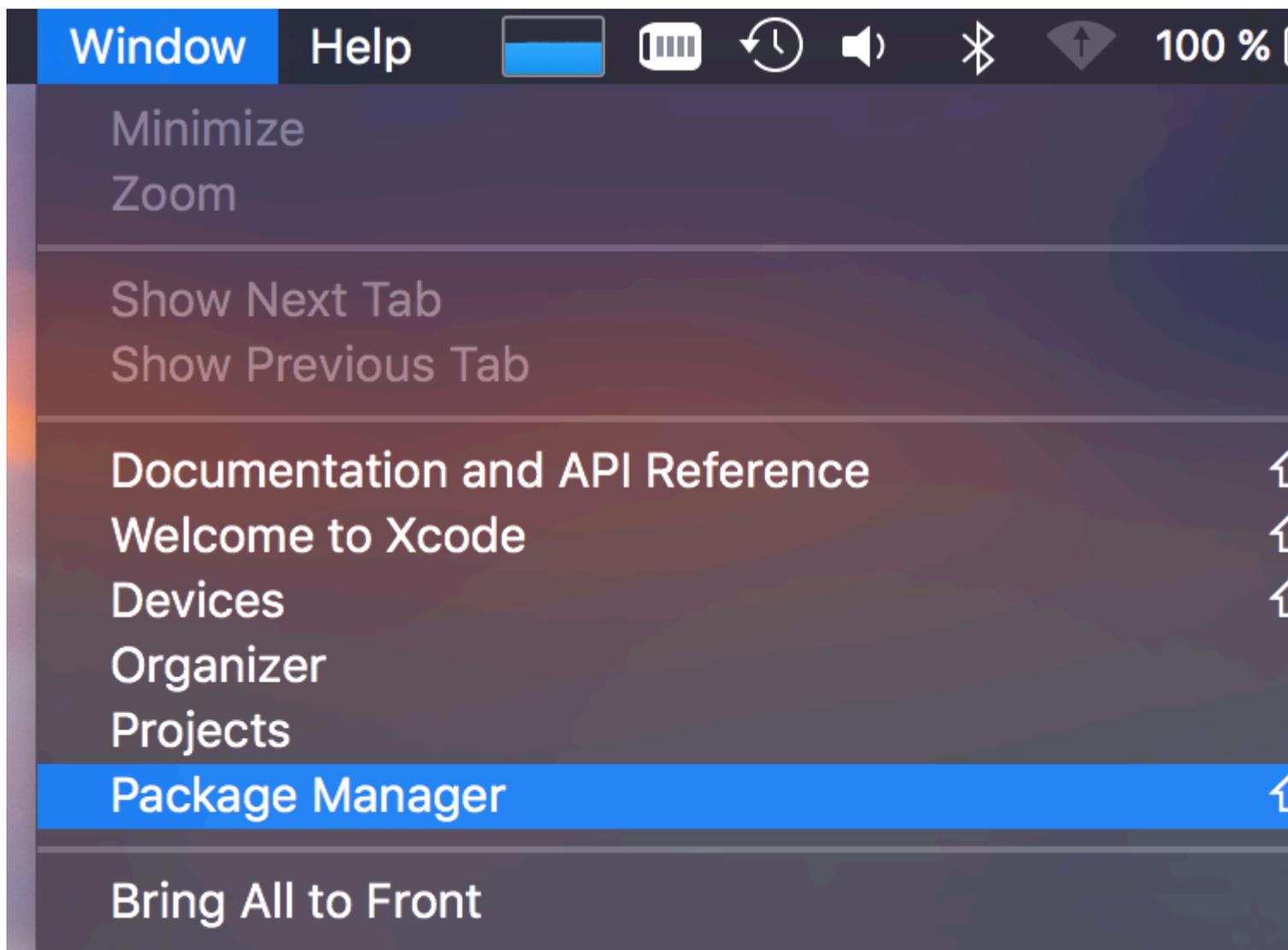
```
1 //
2 // ViewCon
3 // Sample
4 //
5 // Created
6 // Copyrig
7 //
8
9 #import "Vi
10
11 @interface
12
13 @end
14
15 @implementa
16
17 - (void)vie
18     [super
19     // Do a
20 }
21
22 - (void)did
23     [super
24     // Disp
25 }
26
27 - (void)unw
28     subsequ
29     NSLog (@
30
31 @end
```

encima de cualquier método, clase, ... declaración para agregar documentación

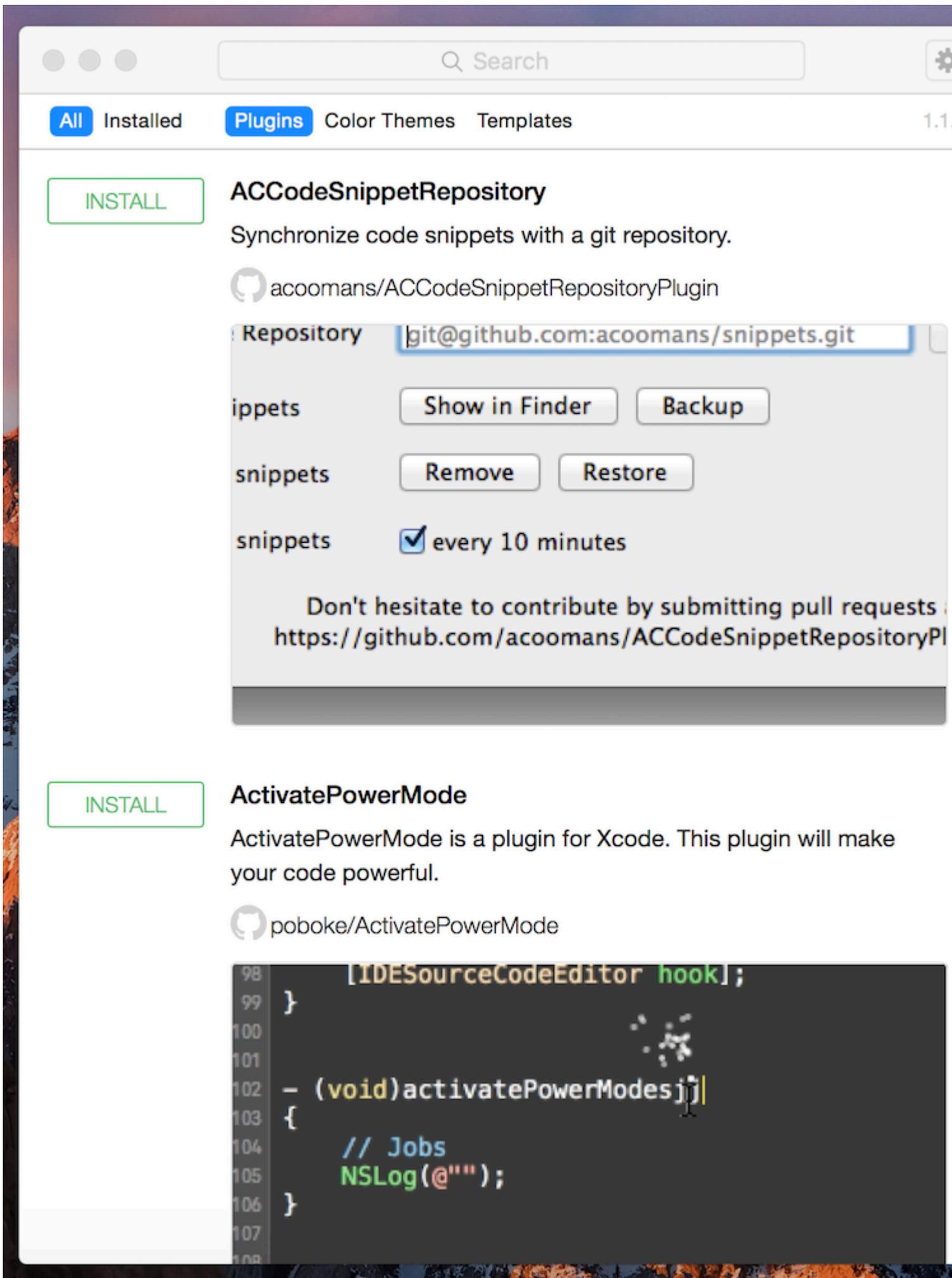
- [XcodeColors](#) - `XcodeColors` consola de color, por ejemplo, utilizando [CocoaLumberjack](#)
- [FuzzyAutocomplete](#) - Escribe "NSLog" y sigue obteniendo `NSLog` autocompletado
- [BuildTimeAnalyzer](#) - Establezca `-Xfrontend -debug-time-function-bodies` bajo `Other Swift flags` en la configuración de construcción y [optimice su tiempo de construcción Swift](#)

Por supuesto, hay muchos más y algunos son muy buenos, Apple ya los implementó en Xcode 8 (FuzzyAutocomplete y VVDocumenter, por ejemplo)

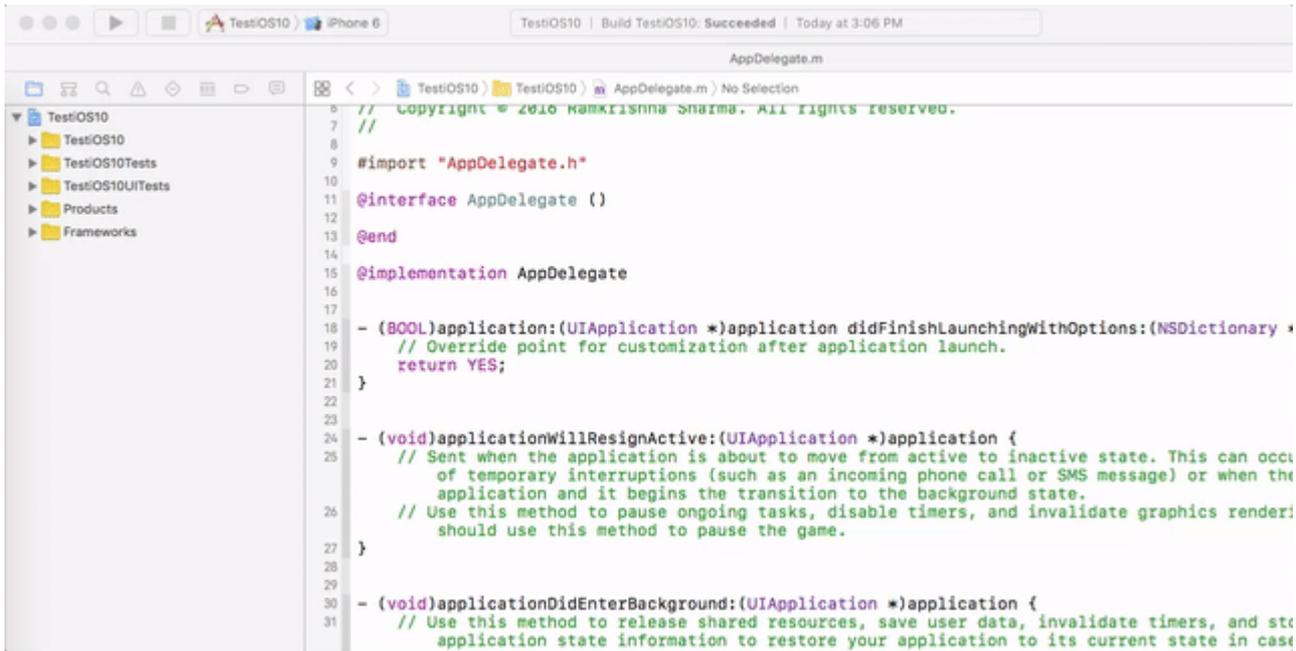
Uso



Presione `⌘ + ⌘ + 9` o use este menú para abrir el Administrador de paquetes.



2. En sus variables de entorno, establezca OS_ACTIVITY_MODE = deshabilitar

A screenshot of the Xcode IDE interface. The top status bar shows 'TestiOS10 | Build TestiOS10: Succeeded | Today at 3:06 PM'. The main editor window displays the AppDelegate.m file. The code includes a copyright notice, an import statement for AppDelegate.h, an @interface AppDelegate {}, an @end, and an @implementation AppDelegate. Three methods are implemented: application:didFinishLaunchingWithOptions:, applicationWillResignActive:, and applicationDidEnterBackground:.

```
0 // Copyright © 2016 Kankrishna Sharma. All rights reserved.
1 //
2
3 #import "AppDelegate.h"
4
5 @interface AppDelegate {}
6
7 @end
8
9 @implementation AppDelegate
10
11 - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
12     // Override point for customization after application launch.
13     return YES;
14 }
15
16 - (void)applicationWillResignActive:(UIApplication *)application {
17     // Sent when the application is about to move from active to inactive state. This can occur
18     // due to temporary interruptions (such as an incoming phone call or SMS message) or when the
19     // application is terminated. Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering.
20     // Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering.
21     // Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering.
22     // Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering.
23 }
24
25 - (void)applicationDidEnterBackground:(UIApplication *)application {
26     // Use this method to release shared resources, save user data, invalidate timers, and store
27     // application state information to restore your application to its current state in case
28     // the application is terminated.
29 }
30
31
```

Lea Consejos de Xcode en línea: <https://riptutorial.com/es/xcode/topic/3349/consejos-de-xcode>

Capítulo 5: Creando controles personalizados en Interface Builder con @IBDesignable

Observaciones

Se hizo mucho más fácil crear controles personalizados en Interface Builder con la introducción de las directivas `@IBDesignable` y `@IBInspectable` en Swift. Los desarrolladores ahora pueden crear controles ricos, complejos y totalmente animados utilizando solo unas pocas líneas adicionales de código. Estoy sorprendido por la cantidad de desarrolladores que aún no han adoptado esta función, y frecuentemente encuentro que agregar solo algunas líneas de código a las clases existentes puede hacer que sea mucho más fácil trabajar con ellas.

Tenga en cuenta que estas características también están disponibles en Objective-C y son una excelente manera de dar vida a las clases anteriores. Los equivalentes sintácticos en Objective-C son `IB_DESIGNABLE` e `IBInspectable`, pero por ahora me concentraré en ejemplos en Swift.

Examples

Una vista redondeada en vivo

Este es un requisito tan común en el desarrollo de iOS, y siempre fue algo que tenía que hacerse puramente en código (o usando imágenes, ¡puaj!). Ahora es increíblemente fácil obtener una vista previa de este tipo de cosas en Interface Builder, no hay absolutamente ninguna excusa para no usarlo.

Aquí está el código: -

```
import UIKit

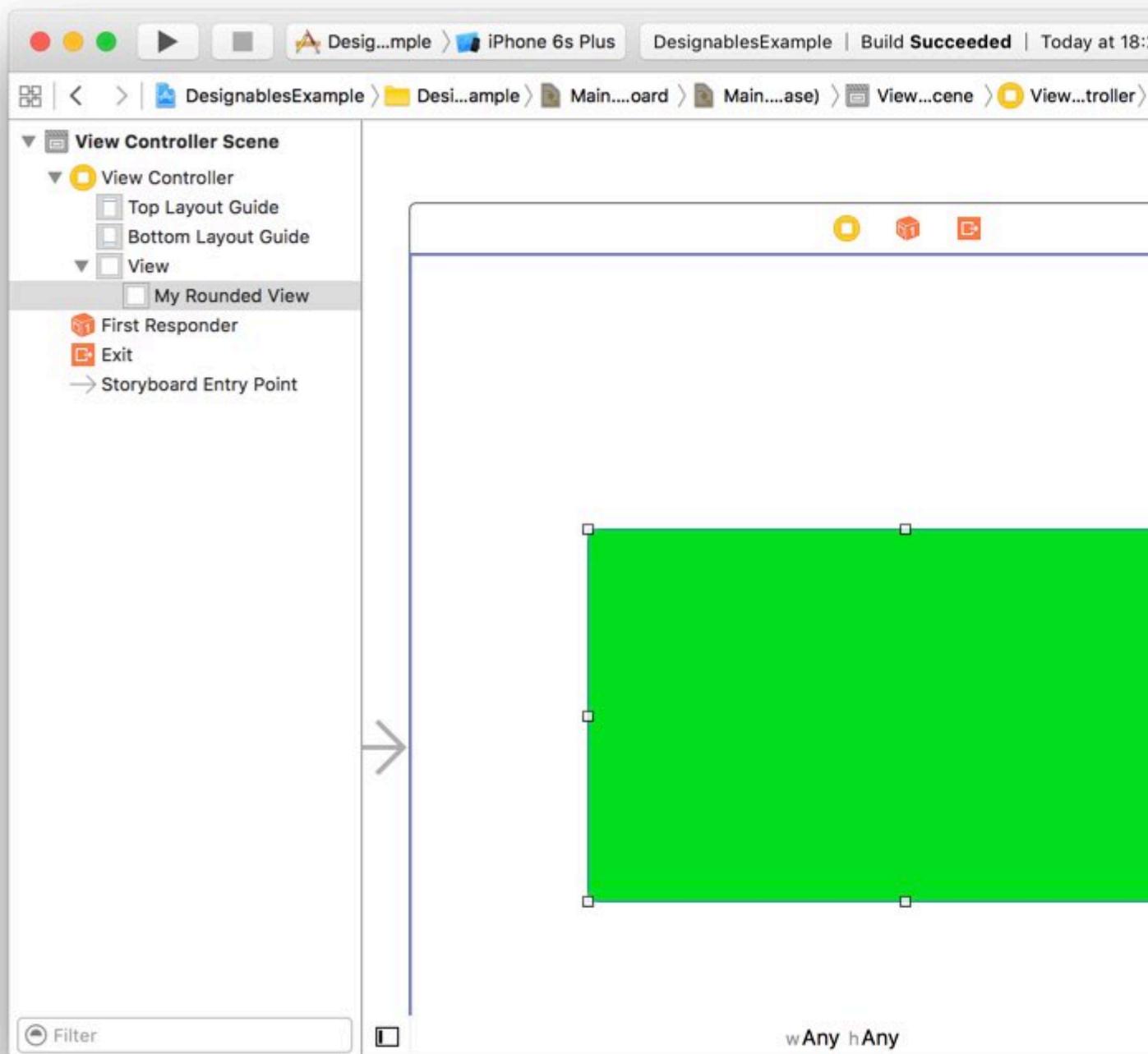
@IBDesignable
class MyRoundedView: UIView {

    @IBInspectable var radius: CGFloat = 8 {
        didSet {
            self.layer.cornerRadius = radius
        }
    }

    override func awakeFromNib() {
        self.layer.cornerRadius = self.radius
        self.layer.masksToBounds = true
    }
}
```

Para usar esta clase, agréguela a su proyecto y luego abra el guión gráfico en IB y cree una vista

normal de un tamaño decente. Asígnale un color de fondo para que pueda verlo, luego navegue hasta el Inspector de identidad en el panel Utilidades de la derecha y cambie el tipo de clase en el menú desplegable a `MyRoundedView`.

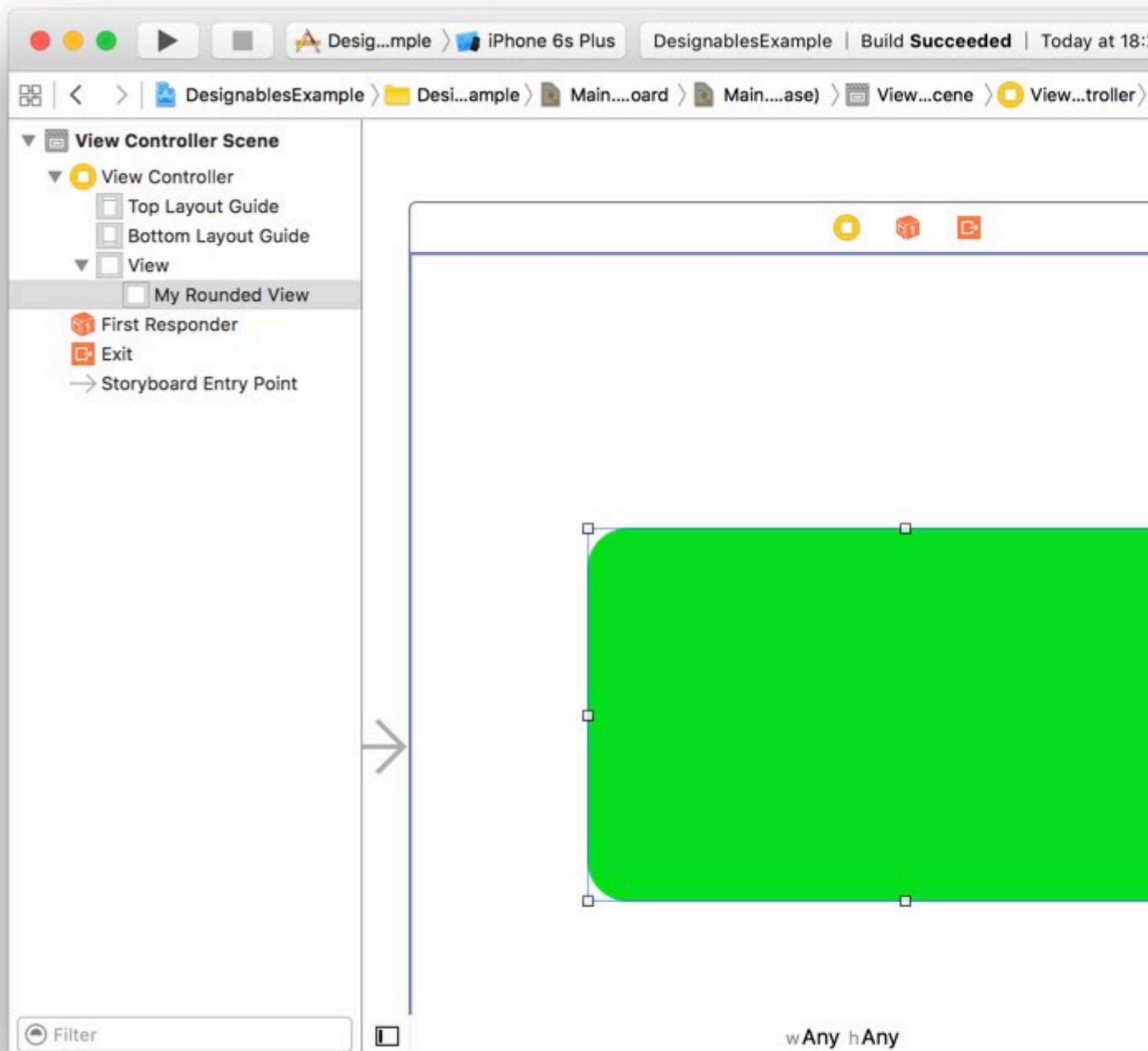


Al hacer esto, debería ver una tercera etiqueta debajo de "Clase" y "Módulo" que dice "Designables", y debería decir "Actualización" por un momento antes de cambiar a "Actual". Esto significa que Xcode ha recompilado su código para `MyRoundedView` éxito.

Ahora puede abrir el Inspector de atributos y debería ver (quizás después de una breve pausa) una nueva sección en la parte superior del panel con el encabezado "Mi vista redondeada" y un nuevo atributo etiquetado "Radio" con el valor 8 (porque es el valor inicial que establecemos en el

código). Esto apareció en el inspector de atributos porque lo marcamos como `@IBInspectable`.

¡Ahora puede cambiar esto a otro número y debería ver la actualización del radio de la esquina de la vista redondeada en tiempo real!



Lea [Creando controles personalizados en Interface Builder con @IBDesignable en línea](https://riptutorial.com/es/xcode/topic/6193/creando-controles-personalizados-en-interface-builder-con-ibdesignable):
<https://riptutorial.com/es/xcode/topic/6193/creando-controles-personalizados-en-interface-builder-con-ibdesignable>

Capítulo 6: Depuración

Examples

Puntos de interrupción

En xcode los desarrolladores pueden pausar / interrumpir la ejecución de la aplicación en ejecución y pueden examinar el estado del programa.

Aquí es cómo pausar programas en ejecución:

Simplemente abra cualquier archivo en el que queramos poner un punto de interrupción y haga clic en la línea en el canalón en el lado izquierdo donde queremos pausar la ejecución.

Debug gauges

The screenshot shows the Xcode interface with the following components:

- Debug navigator:** Located on the left, it displays system gauges (CPU at 0%, Memory at 227.6 MB, Disk at 16 KB/s, Network at Zero KB/s) and a thread list. The thread list shows Thread 1 (Queue: com.apple.main-thread) and Thread 84 (Queue: Graph serial queue (serial)). Under Thread 84, several stack frames are listed, including `0 _35-[GraphView _plotAccel...`, `1 _dispatch_call_block_and_r...`, `8 start_wqthread`, and `0 _dispatch_barrier_async_f_slow`.
- Code editor:** The main area shows the implementation of `GraphView`. A breakpoint is set on line 21, and a variable breakpoint is set on line 38. The code includes property declarations for `_routeProvider`, `_graphImage`, `_routeDescription`, `_routeImage`, `_routeStartL`, `_velocityDataLock`, and `_velocityData`. The `_plotAccelerationCurve` method is shown, which uses `dispatch_async` to plot data points on a `UIBezierPath`.
- Debug bar:** Located at the bottom, it shows the current state of the program. The variable breakpoint on line 38 is active, displaying the following values:
 - `self` = (GraphView *) 0x7fe67f6250f0
 - `currentPoint` = (CGPoint) (x=0, y=0)
 - `path` = (UIBezierPath *) 0x7fe67f18a1e0
 - `_graphSerialQueue` = (dispatch_queue_t) 0x7fe67c727e00
 - `velocityDataLock` = (NSLock *) 0x7fe67c727d90

Debug navigator

Breakpoint

Debug bar

Variab

Así que aquí colocamos puntos de interrupción en la línea 21 y 38; cuando la ejecución llega a la

línea 38, Xcode detuvo la ejecución y mostró una línea verde en esa línea.

Debug Gauges nos da un vistazo del uso de la CPU, el uso de la memoria y, en la parte inferior, la pila de ejecución con hilos y nombres de funciones. Podemos saber qué pila o secuencia de funciones llevan la ejecución a esta línea de ruptura.

La vista de variables proporciona todos los detalles de los estados y valores de todas las variables en el alcance de la línea de empanado. Podemos ver sus valores, direcciones de memoria, propiedades de instancias y sus detalles.

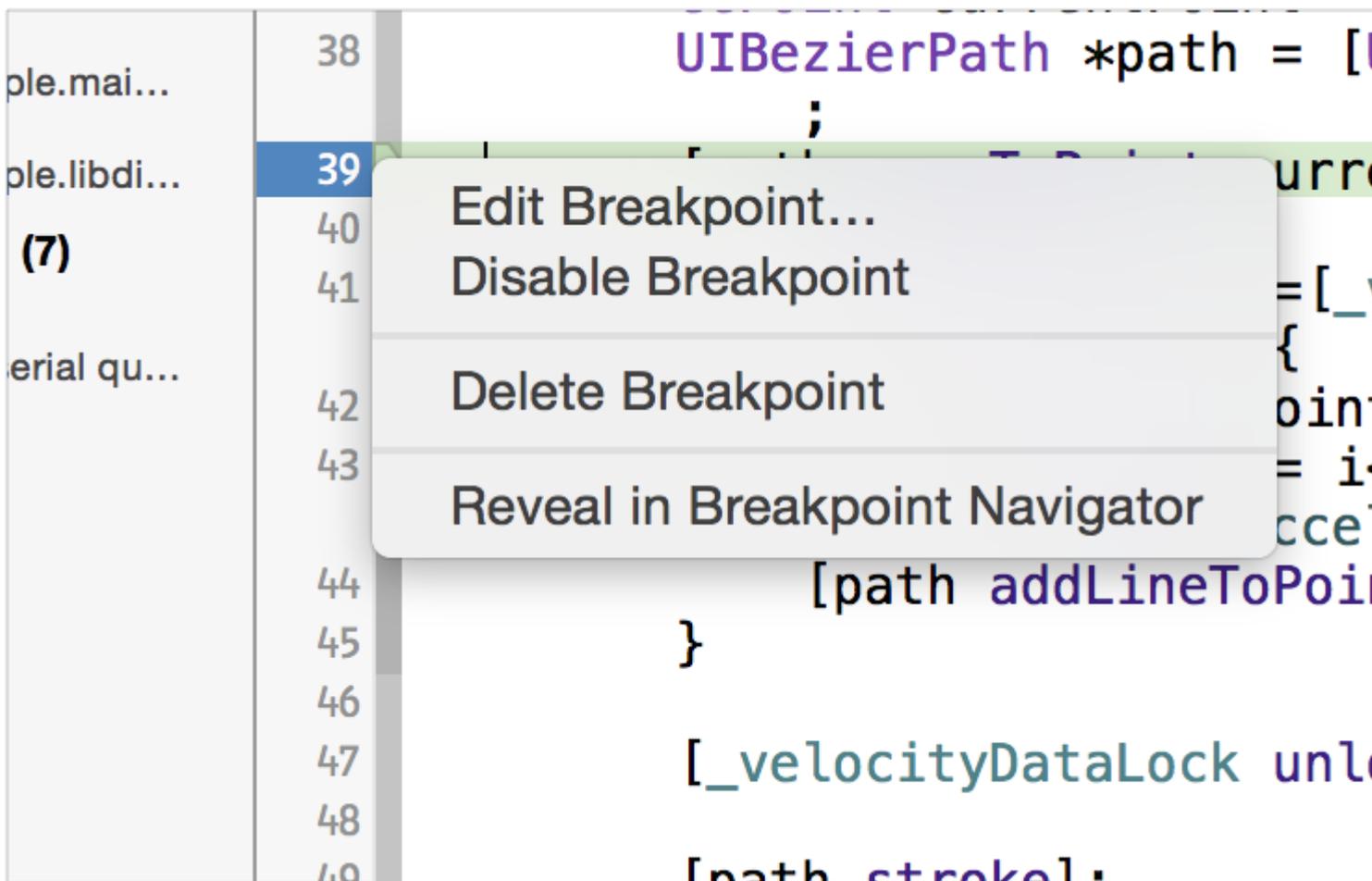
La consola se puede utilizar para imprimir el valor de cualquier variable que esté dentro del alcance. Usando el comando `PO` podemos lograr esto.

Debug Bar tiene controles para puntos de interrupción.

- El primer botón es habilitar / deshabilitar el punto de interrupción pausado.
- Segundo botón utilizado para pausar / reanudar la ejecución de programas.
- El tercero es el botón Paso a paso utilizado para ejecutar la siguiente línea
- Cuarto botón en Step-In usado para ingresar dentro de la función actualmente en ejecución
- El quinto es el botón Salir para salir de la función actual

Configurar punto de interrupción:

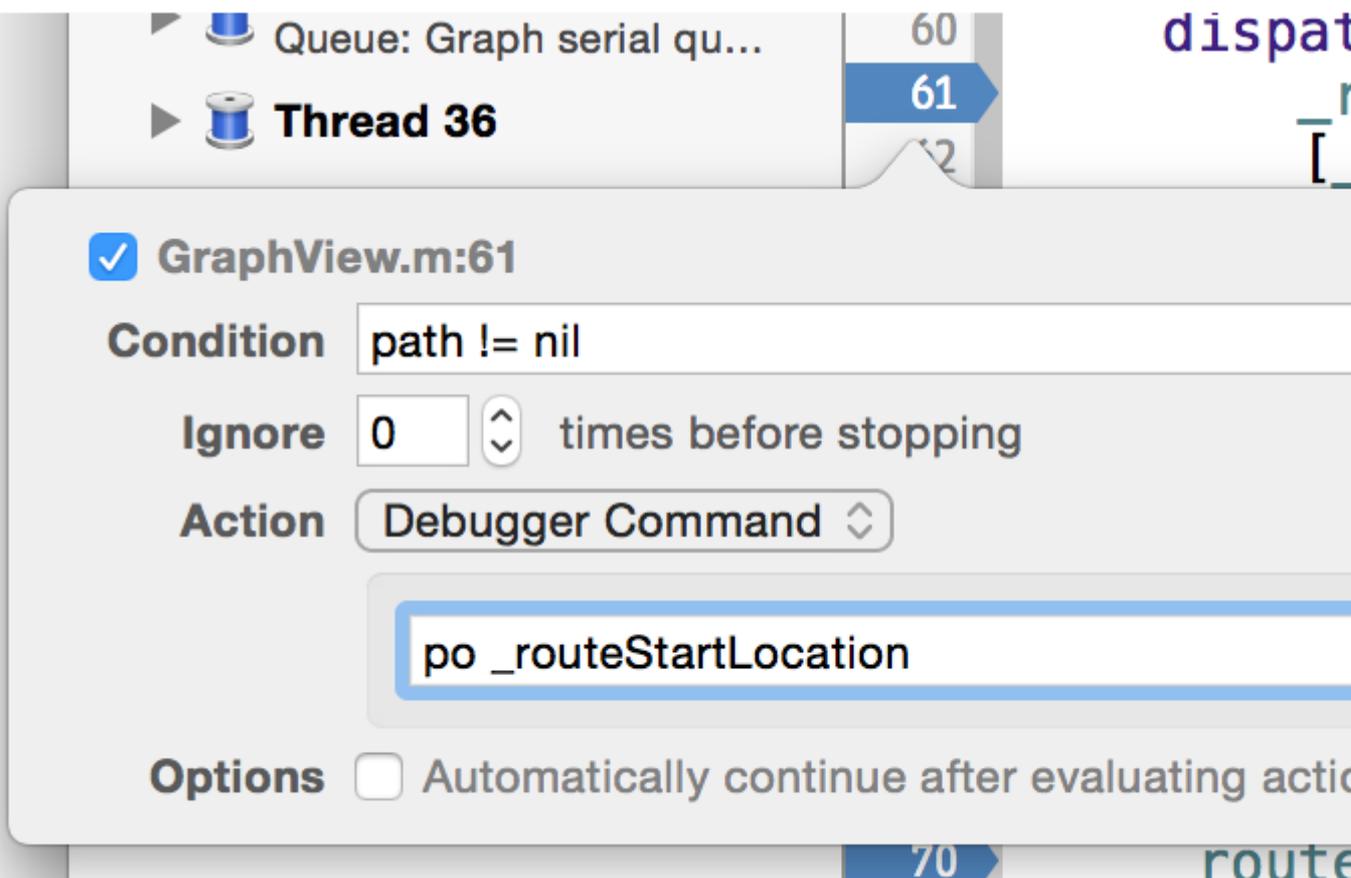
Incluso podemos tener más control sobre los puntos de interrupción.



Eliminar son deshabilitar funciones directas.

Reveal in Navigator nos lleva al navegador de punto de interrupción, donde todos los puntos de corte del proyecto se enumeran como navegador de archivos.

Editar Breakpoint es algo que deberíamos usar más a menudo para una depuración detallada. Podemos configurar puntos de interrupción utilizando esta función. Podemos condicionar y acciones a puntos de interrupción como:



Como se muestra en la imagen, ese punto de interrupción se pausará solo si `path != nil`. Si esta condición es verdadera, entonces la acción `po _routeStartLocation` se ejecuta y mencionó anteriormente `po` imprimirá el valor de `_routeStartLocation` en la consola.

Formulario de explicación detallada, [siga este enlace detallado](#).

Depuración inalámbrica en Xcode-9

Como Apple lanzó recientemente iOS11 y Xcode-9, ahora podemos depurar aplicaciones en dispositivos sin conectar dispositivos a Xcode a través de USB.

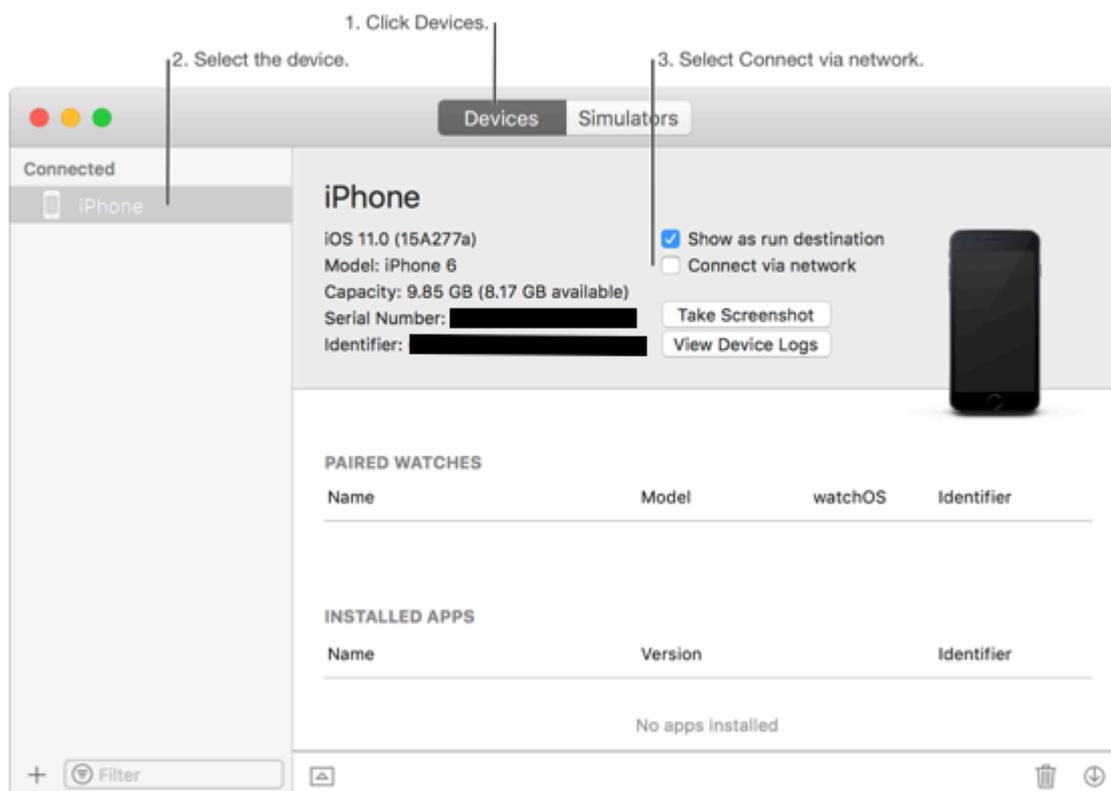
Podemos aprovechar la función de depuración inalámbrica agregada a este Xcode-9.

Para habilitar la depuración inalámbrica, tenemos que configurar algunos pasos en Xcode.

1 Primero conecte el dispositivo que ejecuta iOS11 al Xcode-9.

2 Vaya a Ventana > Dispositivos y simuladores en los menús de Xcode y en la pestaña Dispositivos, se enumerará el dispositivo conectado.

3 Luego marque la casilla de verificación llamada Conectar red como en esta imagen:



(Imagen cortesía de la [publicación SO de Surjeets](#))

4 Luego desconecte su dispositivo del cable USB, asegúrese de que el dispositivo iPhone / iPad / iPod y la Mac con Xcode estén en la misma red inalámbrica.

5 En Xcode verá estos dispositivos en la lista y puede ejecutar directamente su aplicación en ese dispositivo.

Podemos realizar todas las operaciones con Xcode en ese dispositivo como si estuviera conectado mediante USB; excepto que no **podemos ver los registros si la** aplicación se ejecuta con Xcode, póngala en segundo plano y se suspende en estado de fondo y la iniciamos nuevamente. Esto es posible con la depuración USB.

NOTAS:

1 Tenemos que usar Xcode-9, iOS 11 corriendo en el dispositivo

2 Tanto el dispositivo como la Mac deberían estar en la misma red inalámbrica

Lea **Depuración en línea**: <https://riptutorial.com/es/xcode/topic/10459/depuracion>

Capítulo 7: Desarrollo multiplataforma

Examples

TargetConditionals

El encabezado del sistema `TargetConditionals.h` define varias macros que puede usar desde C y Objective-C para determinar qué plataforma está usando.

```
#import <TargetConditionals.h> // imported automatically with Foundation

- (void)doSomethingPlatformSpecific {
#if TARGET_OS_IOS
    // code that is compiled for iPhone / iPhone Simulator
#elif TARGET_OS_MAC && !TARGET_OS_IPHONE
    // code that is compiled for OS X only
#else
    // code that is compiled for other platforms
#endif
}
```

Los valores de las macros son:

7.0

Cuando use iOS 9.1, tvOS 9.0, watchOS 2.0, OS X 10.11 o SDK más recientes:

Macro	Mac	iOS	simulador de iOS	Reloj	Reloj simulador	televisión	Simulador de televisión
TARGET_OS_MAC	1	1	1	1	1	1	1
TARGET_OS_IPHONE	0	1	1	1	1	1	1
TARGET_OS_IOS	0	1	1	0	0	0	0
TARGET_OS_WATCH	0	0	0	1	1	0	0
TARGET_OS_TV	0	0	0	0	0	1	1
TARGET_OS_SIMULATOR	0	0	1	0	1	0	1
TARGET_OS_EMBEDDED	0	1	0	1	0	1	0
TARGET_IPHONE_SIMULATOR	0	0	1	0	1	0	1

7.0

Al usar el iOS 8.4, OS X 10.10 o SDK anteriores:

Macro	Mac	iOS	simulador de iOS
TARGET_OS_MAC	1	1	1
TARGET_OS_IPHONE	0	1	1
TARGET_OS_EMBEDDED	0	1	0
TARGET_IPHONE_SIMULATOR	0	0	1

Lea Desarrollo multiplataforma en línea: <https://riptutorial.com/es/xcode/topic/358/desarrollo-multiplataforma>

Capítulo 8: Herramientas de línea de comandos

Examples

Pruebas de carrera

Para ejecutar sus pruebas de unidad en el simulador usando `xcodebuild use`

Si tiene un espacio de trabajo (por ejemplo, cuando utiliza [CocoaPods](#))

```
xcodebuild \  
-workspace MyApp.xcworkspace \  
-scheme "MyScheme" \  
-sdk iphonesimulator \  
-destination 'platform=iOS Simulator,name=iPhone 6,OS=9.1' \  
test
```

Si tienes un archivo de proyecto

```
xcodebuild \  
-project MyApp.xcproj \  
-scheme "MyScheme" \  
-sdk iphonesimulator \  
-destination 'platform=iOS Simulator,name=iPhone 6,OS=9.1' \  
test
```

Los valores de `destination` alternativos son

```
-destination 'platform=iOS,id=REAL_DEVICE_UDID'  
-destination 'platform=iOS,name=IPHONE NAME'
```

Lista de objetivos disponibles, esquemas y configuraciones de compilación

Para listar todos los esquemas disponibles para el proyecto en su directorio actual

```
xcodebuild -list
```

Opcionalmente, puede pasar una ruta a un proyecto o archivo de área de trabajo

```
xcodebuild -list -workspace ./MyApp.xcworkspace  
xcodebuild -list -project ./MyApp.xcodeproj
```

Ejemplo de salida

```
Information about project "Themoji":  
Targets:
```

```
Themoji
ThemojiUITests
Unit
```

```
Build Configurations:
  Debug
  Release
```

If no build configuration is specified and `-scheme` is not passed then "Release" is used.

```
Schemes:
  Themoji
  ThemojiUITests
  Units
```

Compilar y firmar esquema

Limpeza y compilación de código para iPhone, en el proyecto MyProject para el esquema Qa:

```
xcrun xcodebuild clean \
  -workspace "MyProject.xcworkspace" \
  -scheme "YourScheme" \
  -sdk iphoneos \
  -configuration Debug \
  archive \
  -archivePath builds/MyProject.xcarchive
```

La configuración puede ser `Debug` o `Release` .

Firmando el código previamente compilado:

```
xcrun xcodebuild -exportArchive \
  -archivePath builds/MyProject-Qa.xcarchive \
  -exportOptionsPlist config.plist \
  -exportPath builds
```

`config.plist` contiene la información sobre cómo empaquetar y firmar la aplicación, para el uso de las compilaciones de desarrollo:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>method</key>
  <string>development</string>
  <key>uploadSymbols</key>
  <true/>
</dict>
</plist>
```

Un plist de lanzamiento de App Store debería contener algo como:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>teamID</key>
  <string>xxxxxxxxxxx</string>
  <key>method</key>
  <string>app-store</string>
  <key>uploadSymbols</key>
  <true/>
</dict>
</plist>

```

Donde se puede obtener la identificación del equipo de su llavero.

Todos los parámetros disponibles

- compileBitcode
- embedOnDemandResourcesAssetPacksInBundle
- iCloudContainerEnvironment
- manifest
- method
- onDemandResourcesAssetPacksBaseURL
- teamID
- thinning
- uploadBitcode
- uploadSymbols

Para obtener más información sobre cada uno de los parámetros, ejecute `xcodebuild --help`

Acceda a cualquier herramienta de línea de comandos en el paquete de aplicaciones Xcode (xcrun)

`xcrun` usa la versión Xcode predeterminada del sistema (configurada mediante `xcode-select`) para localizar y ejecutar herramientas de línea de comandos del paquete de aplicaciones Xcode, por ejemplo, `llvm-cov`.

```

# Generate code coverage reports via llvm-cov
# /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin
xcrun llvm-cov [parameters]

# Execute xcodebuild
# /Applications/Xcode.app/Contents/Developer/usr/bin
xcrun xcodebuild [parameters]

# Use Xcode's version of git, e.g., if you have installed a newer version
# /Applications/Xcode.app/Contents/Developer/usr/bin
xcrun git [parameters]

```

Cambio de herramientas de línea de comandos con xcode-select

Imprima la ruta al directorio de desarrollador activo (Xcode seleccionado)

```
xcode-select -p
```

Seleccione una versión diferente de Xcode, por ejemplo, Beta

```
sudo xcode-select -s /Applications/Xcode-beta.app
```

Restablecer la versión predeterminada de Xcode

```
sudo xcode-select -r
```

Esto es equivalente a ejecutar `sudo xcode-select -s /Applications/Xcode.app`

Para más detalles: `man xcode-select`

Lea Herramientas de línea de comandos en línea:

<https://riptutorial.com/es/xcode/topic/2158/herramientas-de-linea-de-comandos>

Capítulo 9: Parques infantiles

Examples

Primeros pasos con Playground

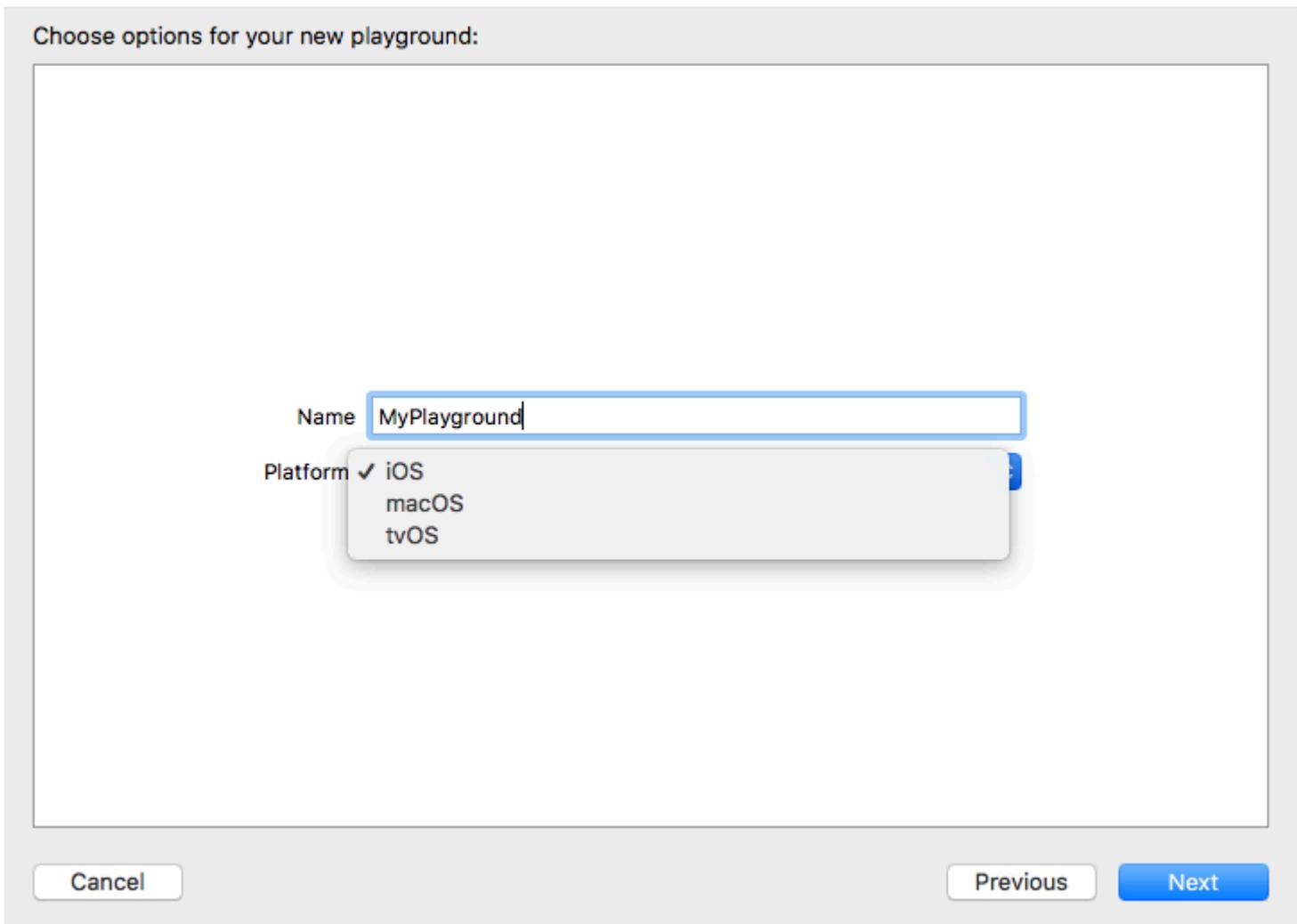
1. Crear un nuevo archivo de juegos:

- Primera opción: desde la pantalla de bienvenida de Xcode, seleccione la primera opción (**Comenzar con un área de juegos**).



- Segunda opción: en el menú, seleccione **Archivo** → **Nuevo** → **Patio de juegos** (N).

2. Nombre su área de juegos y seleccione la plataforma (iOS / macOS / tvOS), luego haga clic en **Siguiente** .



3. En la siguiente pantalla, elija dónde desea guardar su área de juegos, luego haga clic en **Crear** .

Último valor, valor histórico y gráfico

Al usar Playground es fácil ver que sucede dentro de los bucles u objetos mientras ocurre el cambio.

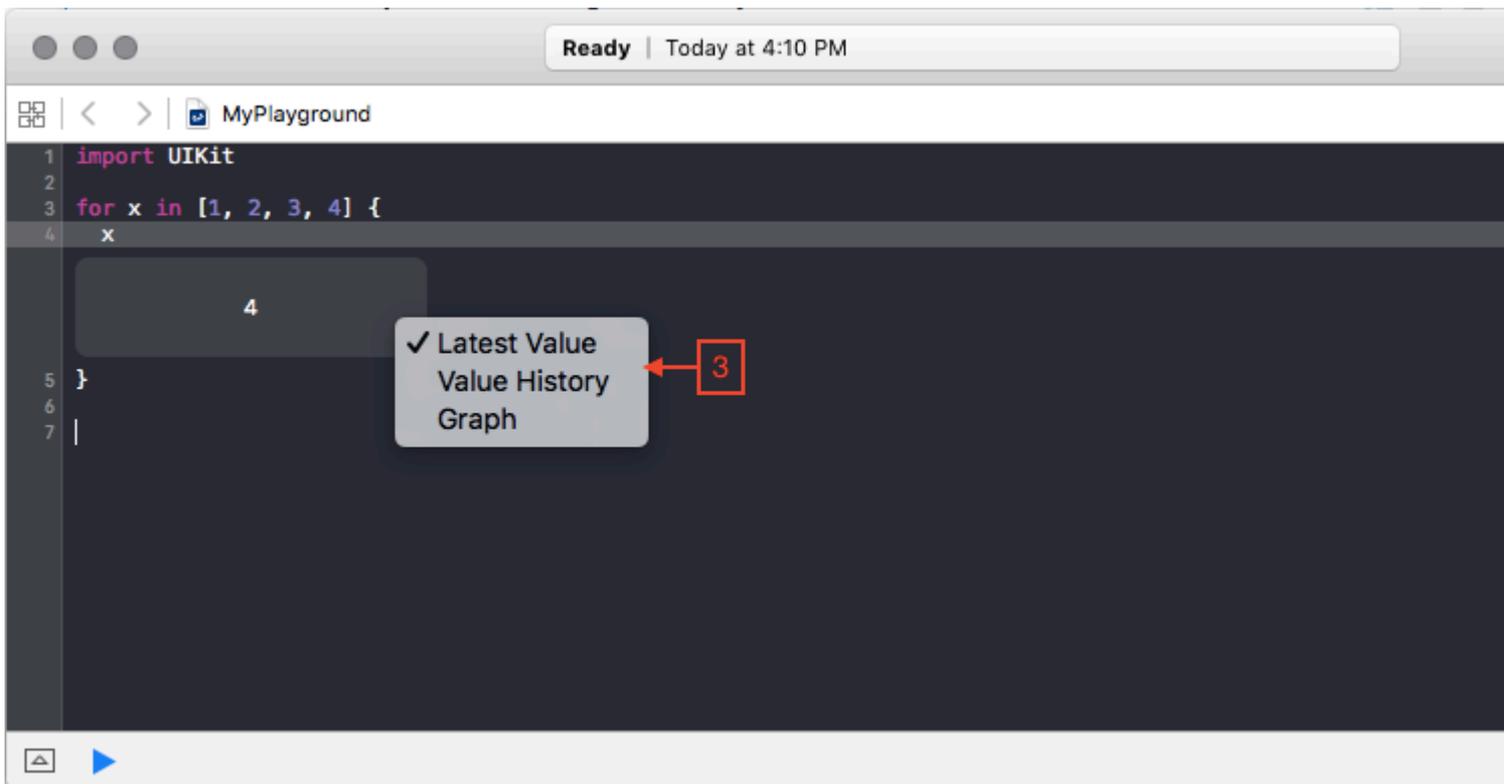
Por ejemplo, en el código a continuación, el valor de `x` cambiará de 1 a 4.

```
import UIKit
for x in [1, 2, 3, 4] {
    x
}
```

(1) Hacer clic en el símbolo del ojo a la derecha nos dará un vistazo rápido.

(2) Al hacer clic en el círculo al lado de él, se abrirá el *último valor* debajo de la línea.

(3) Al hacer clic con el botón derecho en la vista agregada se mostrará un menú desplegable con el **valor más reciente, el historial de valores y el gráfico**.



Adición de imágenes, datos estáticos, sonidos, etc. a un patio de juegos

Imágenes, datos estáticos, sonidos, etc. son recursos en un Patio de Recreo.

1. Si el Navegador de proyectos está oculto, elija Ver> Navegadores> Mostrar Navegador de proyectos (1)
2. Hay varias maneras de agregar archivos
 - Arrastra tus recursos a la carpeta de `Resources` o
 - Seleccione la carpeta Recursos y elija Archivo> Agregar archivos a "Recursos" o
 - control-clic en la carpeta Recursos y elija Agregar archivos a "Recursos"
3. Usa tu recurso. Por ejemplo, `let i = UIImage(named: "tacos.jpg")`

Lea Parques infantiles en línea: <https://riptutorial.com/es/xcode/topic/1236/parques-infantiles>

Capítulo 10: Personalizando Xcode IDE

Introducción

Esta es una colección de diferentes consejos y trucos para personalizar y mejorar su IDE de Xcode.

Examples

Abrir Terminal en la carpeta del proyecto Xcode actual

Xcode tiene capacidad para ejecutar cualquier script con teclas de acceso rápido.

Aquí hay un ejemplo de cómo asignar las teclas de `⌘+⌘+^+⌘+T` rápido `⌘+⌘+^+⌘+T` para abrir la aplicación Terminal en la carpeta del proyecto actual.

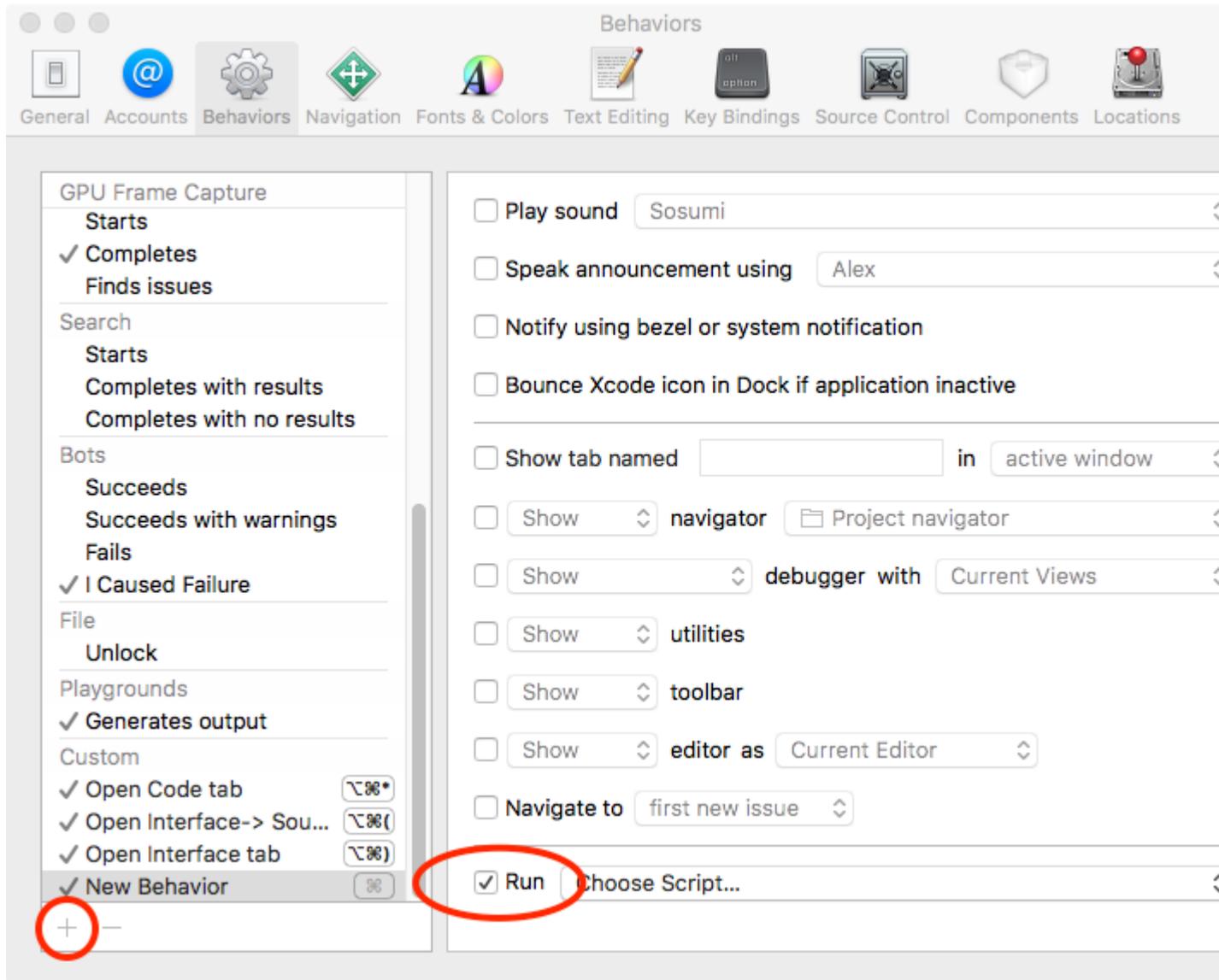
1. Crea un script bash y guárdalo en alguna carpeta.

```
#!/bin/bash

# Project Name:  $XcodeProject
# Project Dir:   $XcodeProjectPath
# Workspace Dir: $XcodeWorkspacePath

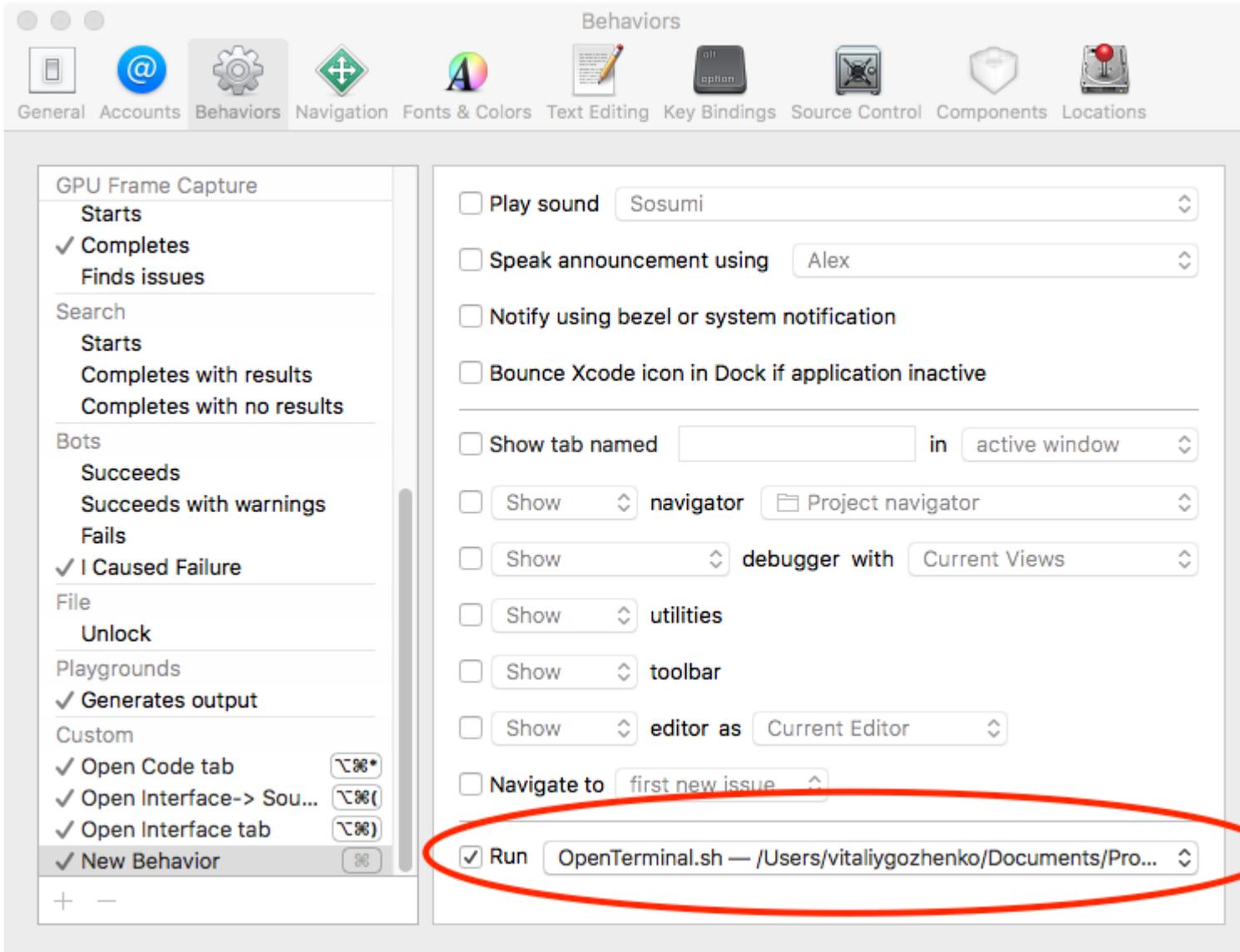
open -a Terminal "$(dirname $XcodeProjectPath)"
```

2. Haga ejecutable el script: abra la Terminal en la carpeta del script y ejecute `chmod +x your_script_name.sh`
3. Abrir Preferencias de Xcode en la pestaña Comportamientos
4. Agrega un nuevo comportamiento personalizado tocando `+` en la esquina inferior izquierda
5. Verifique la acción `Run` a la derecha

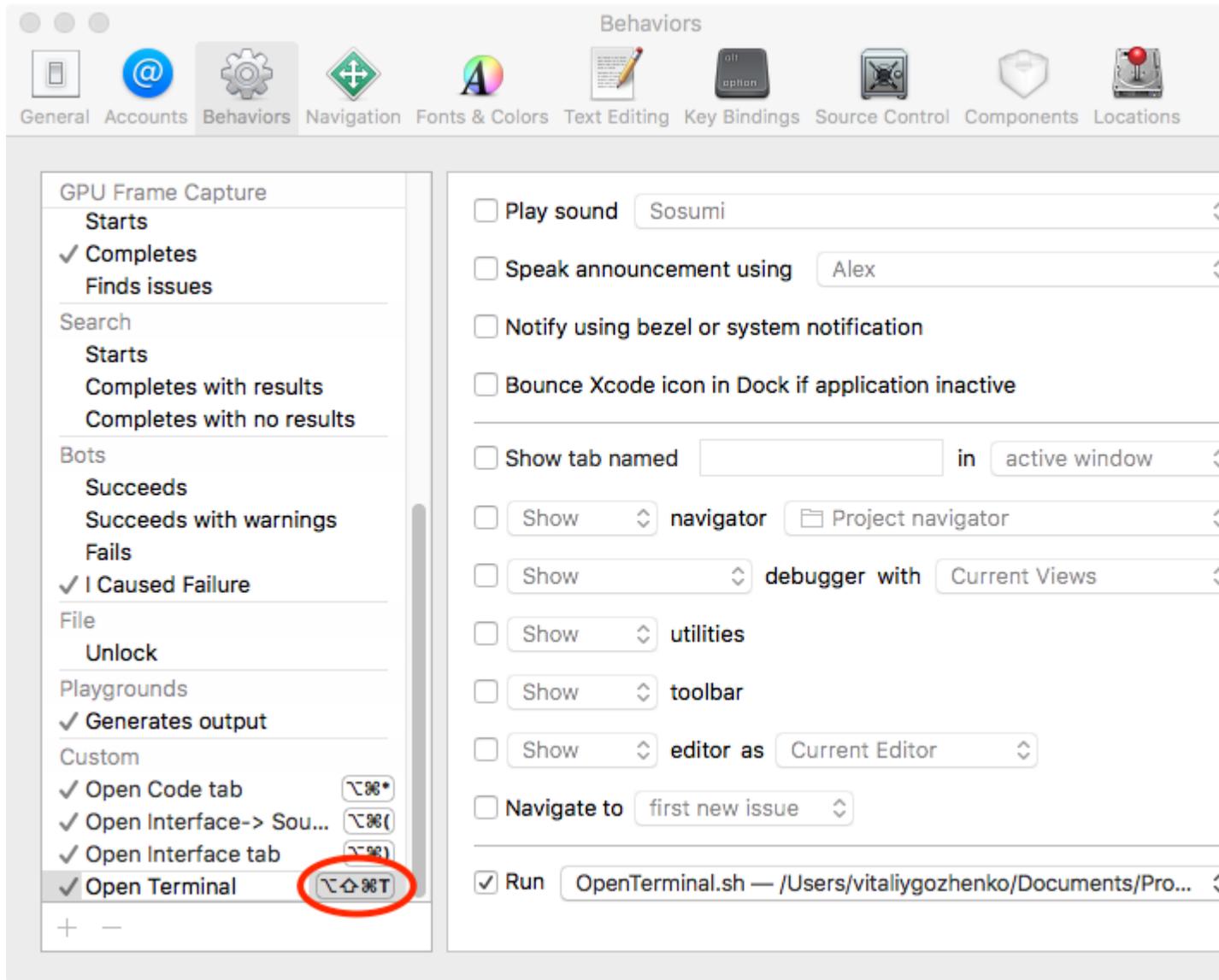


6. Elija el script, que creará previamente haciendo clic en `Choose Script...` dos veces.

Si su script está en gris, asegúrese de ejecutar `chmod +x` en su archivo de script



7. Asigne teclas de acceso rápido (por ejemplo, $\text{⌘} + \text{⌘} + \text{⌘} + \text{⌘} + \text{T}$) a su comportamiento y cambie su nombre



Ahora puede abrir el terminal en la carpeta del proyecto con una tecla de acceso directo.

Este es solo un ejemplo del uso de comportamientos de Xcode, pero puede crear cualquier script e iniciar cualquier aplicación con él.

Autor del script Bash: <http://mattorb.com/xcode-behaviors-for-fun-and-profit/>

Borrar datos derivados con teclas de acceso rápido

De la misma manera que en `Open Terminal in current Xcode project folder` ejemplo de la `Open Terminal in current Xcode project folder`, puede agregar carpetas de datos derivadas con teclas de acceso rápido.

Cree un comportamiento personalizado (siga los pasos en [Abrir Terminal en la carpeta de proyecto actual de Xcode](#)). Pero usa otro script.

Texto de guión:

```
#!/bin/bash
```

```
rm -rf $HOME"/Library/Developer/Xcode/DerivedData/"
```

Lea Personalizando Xcode IDE en línea:

<https://riptutorial.com/es/xcode/topic/8260/personalizando-xcode-ide>

Capítulo 11: Proyectos y espacios de trabajo

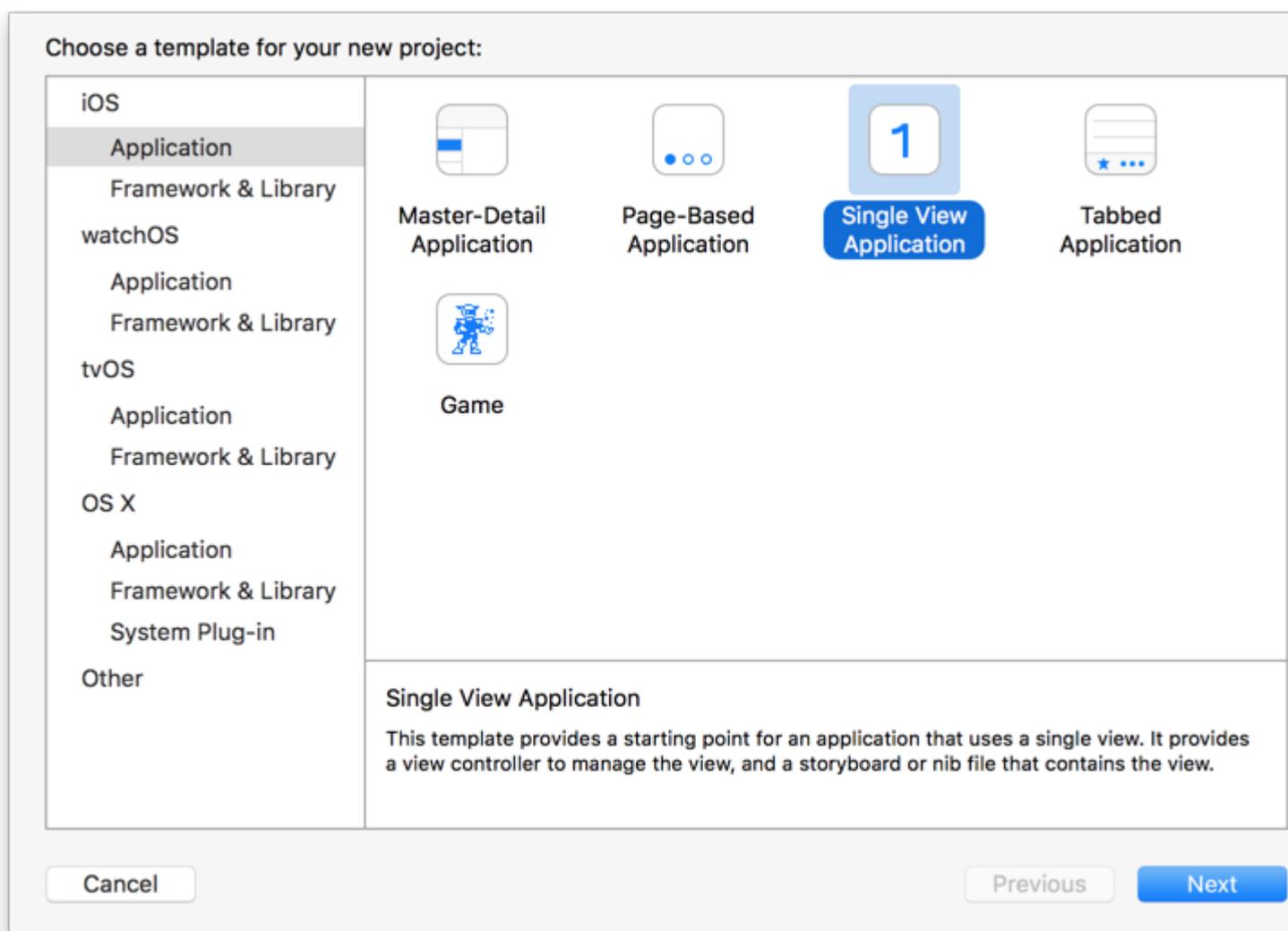
Examples

Resumen de proyectos

Los **proyectos de** Xcode se utilizan para organizar archivos de origen, dependencias de biblioteca y otros recursos, así como la configuración y los pasos necesarios para construir los productos del proyecto. **Los espacios de trabajo** son grupos de proyectos y otros archivos.

Crear un proyecto

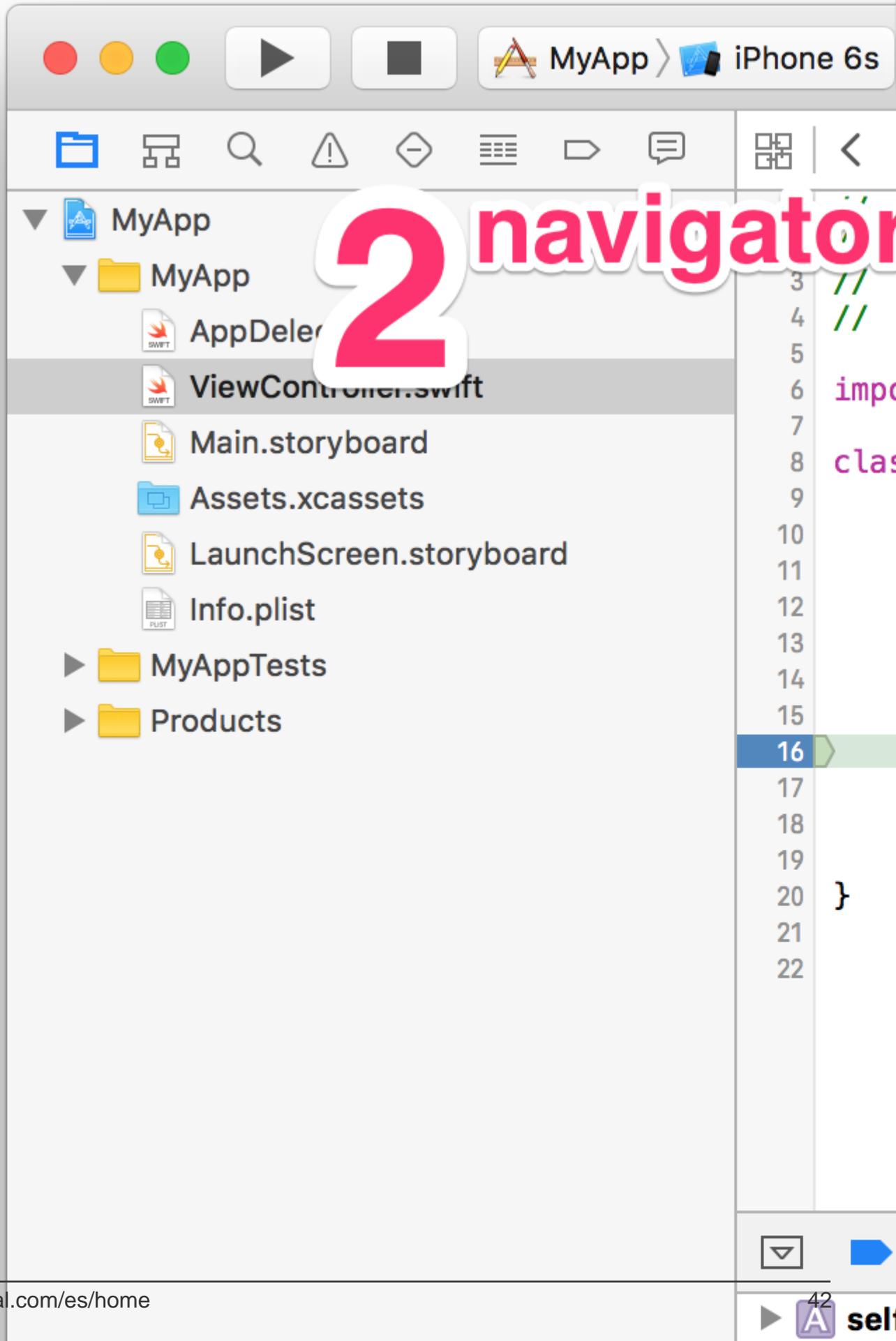
Puede crear un **Nuevo Proyecto** (N) a partir de una serie de plantillas integradas:



Trabajando con proyectos

Una ventana de proyecto de Xcode incluye:

1. **Barra de herramientas** (arriba)
2. **Navegante** (izquierda)
3. **Editor** (centro)
4. **Inspector** (derecha)
5. **Vista de variables** (inferior media central izquierda)
6. **Salida de consola** (inferior derecha media)
7. **Biblioteca** (abajo a la derecha)

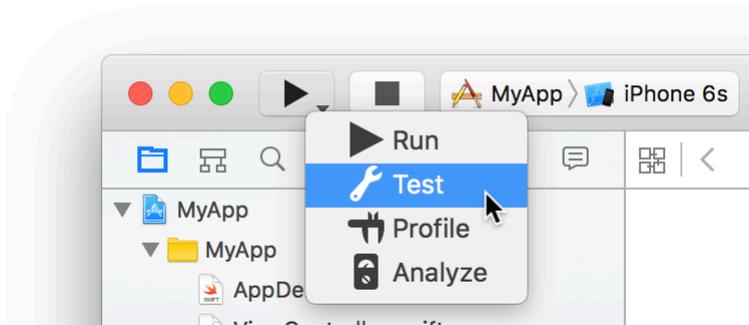


2 navigator

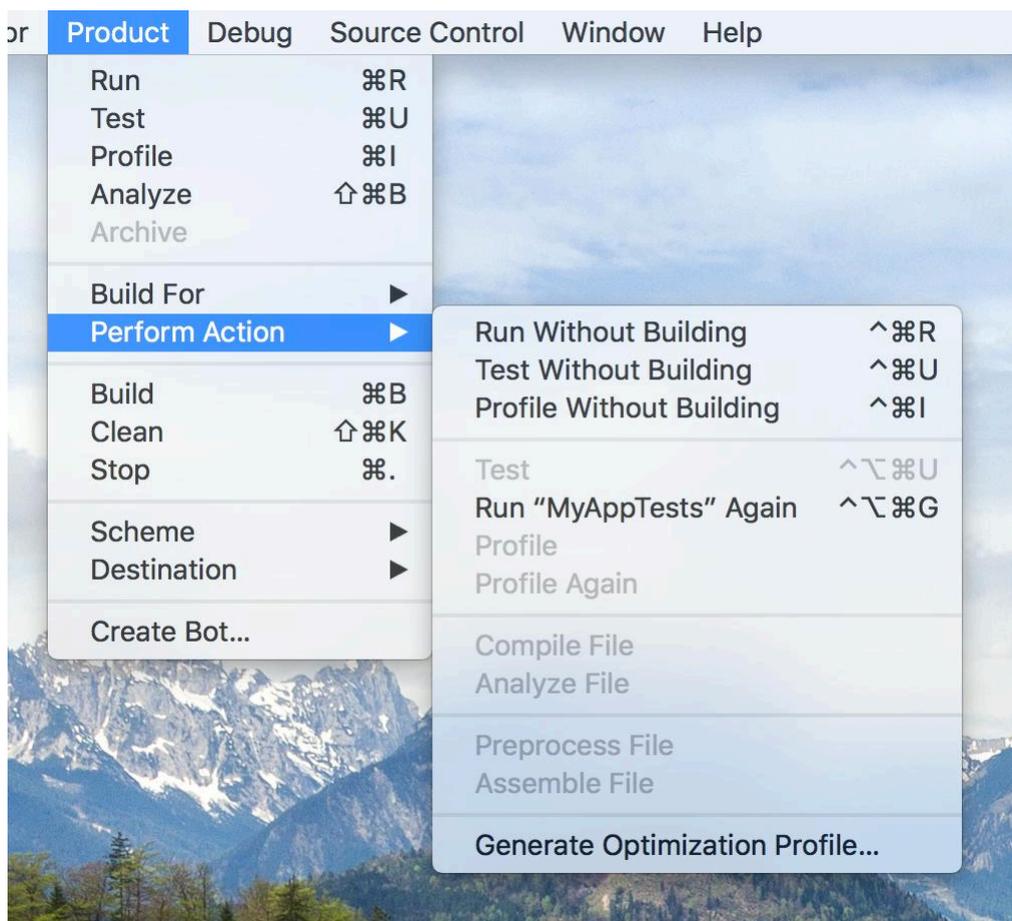
R) para compilar y ejecutar su proyecto. Haga clic en Detener (o presione .) Para detener la ejecución.



Haga clic y mantenga presionado para ver las otras acciones, Prueba (U), Perfil (I) y Analizar (B). Mantenga presionadas las teclas modificadoras `⌘ option`, `⇧ shift`, y `⌘ ^ control` para más variantes.



Las mismas acciones están disponibles en el menú Producto:



Ajuste el espacio de trabajo a sus necesidades y navegue libremente por él

Una de las cosas que realmente puede aumentar su productividad mientras escribe el código es

navegar efectivamente por el área de trabajo. Esto también significa hacerlo cómodo por el momento. Es posible lograr esto ajustando las áreas de áreas de trabajo que ve.

Los botones en la parte superior de las áreas de navegación y asistente no son tan grandes y son un poco difíciles de hacer clic con el puntero del mouse. Es por eso que hay atajos útiles y fáciles de recordar que le permiten cambiar entre diferentes pestañas u ocultar el área por completo.

Se puede cambiar entre los diferentes paneles en el área de navegación manteniendo pulsado el botón `⌘` (Comando) y presionando en diferentes teclas de números del 1 al 8 ó 0. La clave `⌘` alterna la presencia del navegador. Aquí hay una lista de atajos para su conveniencia:

1. `⌘ + 1` - Navegador de archivos;
2. `⌘ + 2` - navegador de símbolos;
3. `⌘ + 2` - Búsqueda (también accesible a través de `⌘ + ⌘ + F`);
4. `⌘ + 4` - Advertencias, errores y mensajes de análisis estático;
5. `⌘ + 5` - Pruebas disponibles en el proyecto;
6. `⌘ + 6` - Panel de sesión de depuración;
7. `⌘ + 7` - Puntos de interrupción;
8. `⌘ + 8` - Navegador de informes / historial de acciones;
9. `⌘ + 0` - Mostrar / Ocultar el panel del navegador;

Puede cambiar entre los diferentes paneles en el área del inspector manteniendo presionados `⌘` y `⌘` y presionando diferentes teclas de dígitos del 1 al 6 (dependiendo del editor activo actual: ya sea código, guión gráfico, xib u otro tipo de recurso). Si presiona `⌘` mientras mantiene estos dos botones, se ocultará el área del inspector.

Entonces, si está editando un guión gráfico y necesita más espacio visible, solo toque `⌘ + 0` y `⌘ + ⌘ + 0` para obtener píxeles adicionales para el lienzo.

Mientras que la conmutación paneles a cada lado en su mayoría depende de la combinación de `⌘` o `⌘` y `⌘`, los campos de búsqueda inferiores se activan mediante la celebración de `⌘` y `⌘` y presionando ya sea `⌘` para el área de la barra de navegación de búsqueda o `⌘` para la biblioteca barra de búsqueda zona.

La activación del área de búsqueda de navegación puede ayudarlo a reducir la lista de elementos que se muestran en el área del navegador según el navegador que esté activo (filtre los archivos en el navegador de archivos, los símbolos en el navegador de símbolos, las pruebas en el navegador de prueba, etc.).

La activación del panel de inspección le ayudará a filtrar la lista de plantillas de archivos, fragmentos de código, objetos o recursos multimedia. Intente usar este campo de búsqueda cuando tenga el guión gráfico abierto y rápidamente necesita encontrar un componente

```
UINavigationController UITableViewCell .
```

Hablando de la biblioteca, puede cambiar entre los paneles de la biblioteca (plantillas de archivos, fragmentos de código, bibliotecas de objetos y multimedia) `⌘ + ⌘ + ⌘` y las teclas de dígitos correspondientes: 1 a 4.

Lea Proyectos y espacios de trabajo en línea: <https://riptutorial.com/es/xcode/topic/335/proyectos->

Creditos

S. No	Capítulos	Contributors
1	Empezando con Xcode	Anh Pham , Community , Jbryson , jtbandes , Md. Ibrahim Hassan , Undo
2	Características de Xcode 8	Rashwan L , Sharukh Mastan
3	Certificados, perfiles de aprovisionamiento y firma de código	KrauseFx
4	Consejos de Xcode	Anand Prem , Finn Gaida , jtbandes , user459460
5	Creando controles personalizados en Interface Builder con @IBDesignable	Echelon
6	Depuración	D4ttatraya
7	Desarrollo multiplataforma	J F , jtbandes
8	Herramientas de línea de comandos	Ali Beadle , David Snabel-Caunt , Fabio , Idan , Jens Meder , KrauseFx
9	Parques infantiles	Anh Pham , Idan , Rob Wright
10	Personalizando Xcode IDE	Vitaliy Gozhenko
11	Proyectos y espacios de trabajo	Eimantas , jtbandes