



Existen muchas maneras de instalar Flutter en Mac, la mayoría usando **el navegador, el Finder y Drag & Drop**.

## Tabla de contenidos

1. Requisitos para instalar Flutter en Mac.
2. ¿Cómo instalar Flutter en Mac?
3. Instala Xcode
4. Configura los command-line tools
5. Instalar Flutter en Mac.
  - 5.1. Descargar Flutter en Mac.
  - 5.2. Descomprimir Flutter.
  - 5.3. Hacer que el comando flutter sea global.
6. ¡Hagamos una prueba!
7. Instala brew.
8. Visual Studio Code.
  - 8.1. Extensiones de Visual Studio Code.
9. Instalar Android Studio.
  - 9.1. Primer inicio e Instalar un SDK
  - 9.2. Instalar el SDK de Android.
10. Configurar un emulador de Android.
11. Últimas anotaciones.



Si eres un novato en el **desarrollo de software y aplicaciones**, y la terminal te asusta, te recomiendo que sigas una de esas maneras. Pero si lo que quieres es tener tu entorno de desarrollo lo más rápido posible y te sientes cómodo trabajando en la terminal de Mac.

Te voy a explicar cómo tener tu entorno de desarrollo lo más rápido y liviano posible. Claro, tan liviano como Mac permite, puesto que las herramientas que vas a instalar de por sí son bastante pesadas.

Antes de que empecemos, una advertencia. Vas a descargar más de 4 gigas en solo instaladores y necesitarás unos 30 gigas de espacio en tu disco duro para que todo esto pueda funcionar.

Si no te interesa entender que estamos haciendo y solo quieres tener todas las aplicaciones instaladas. Solo tienes que Copiar y Pegar los comandos.

## Requisitos para instalar Flutter en Mac.

Los requisitos son pocos. Al fin y al cabo lo que vamos a instalar son los requisitos para que puedas empezar a desarrollar.

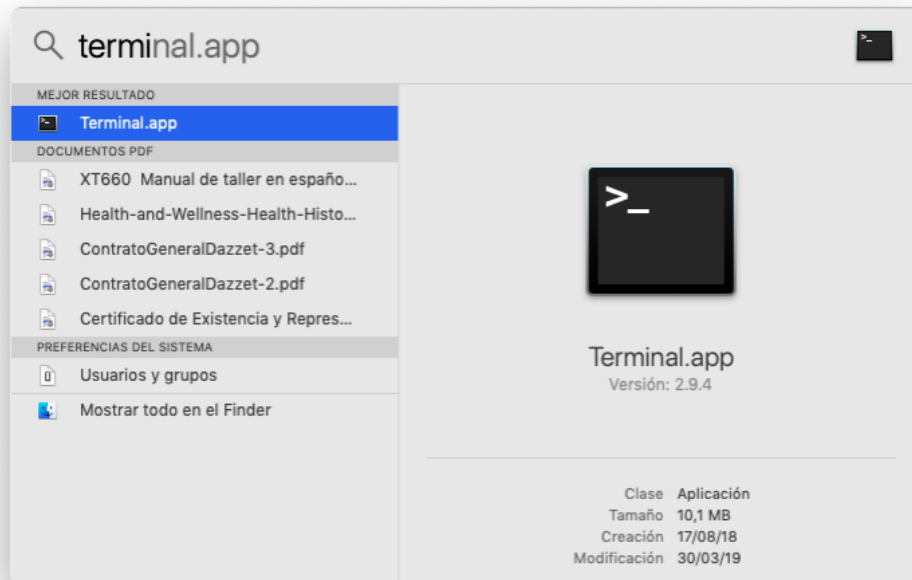
Para seguir este artículo vas a necesitar:

Una Mac con la última versión de macOS (Yo tengo Mojave)





Una aplicación de terminal instalada. Ya macOS viene con **Terminal** pero yo voy a usar **iTerm**.



Un par de horas disponibles. En serio...  
¡vas a descargar mucha información!

Ok, entonces abre tu terminal y empecemos a instalar Flutter en Mac.



## ¿Cómo instalar Flutter en Mac?

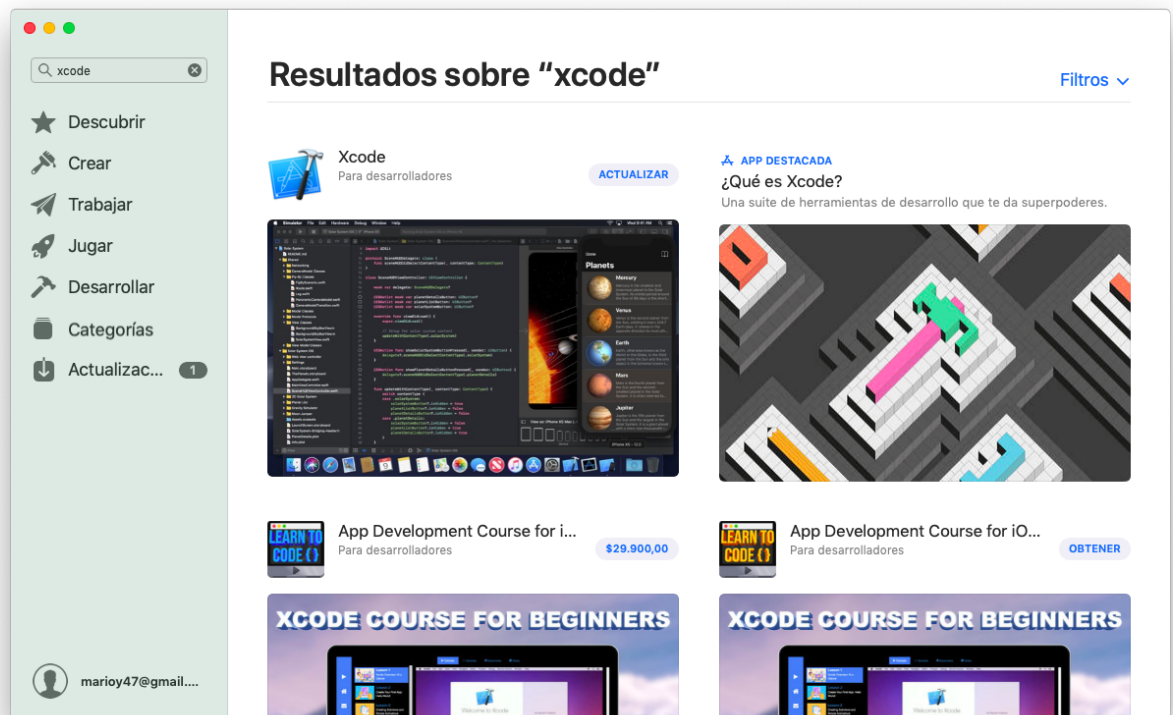
### Instala Xcode

Este es uno de los pocos pasos que debes hacer por fuera de la terminal, ya que para instalar Xcode es necesario que lo hagas desde la aplicación *App Store* de Apple.

Puedes iniciar el *App Store* desde la terminal usando el comando

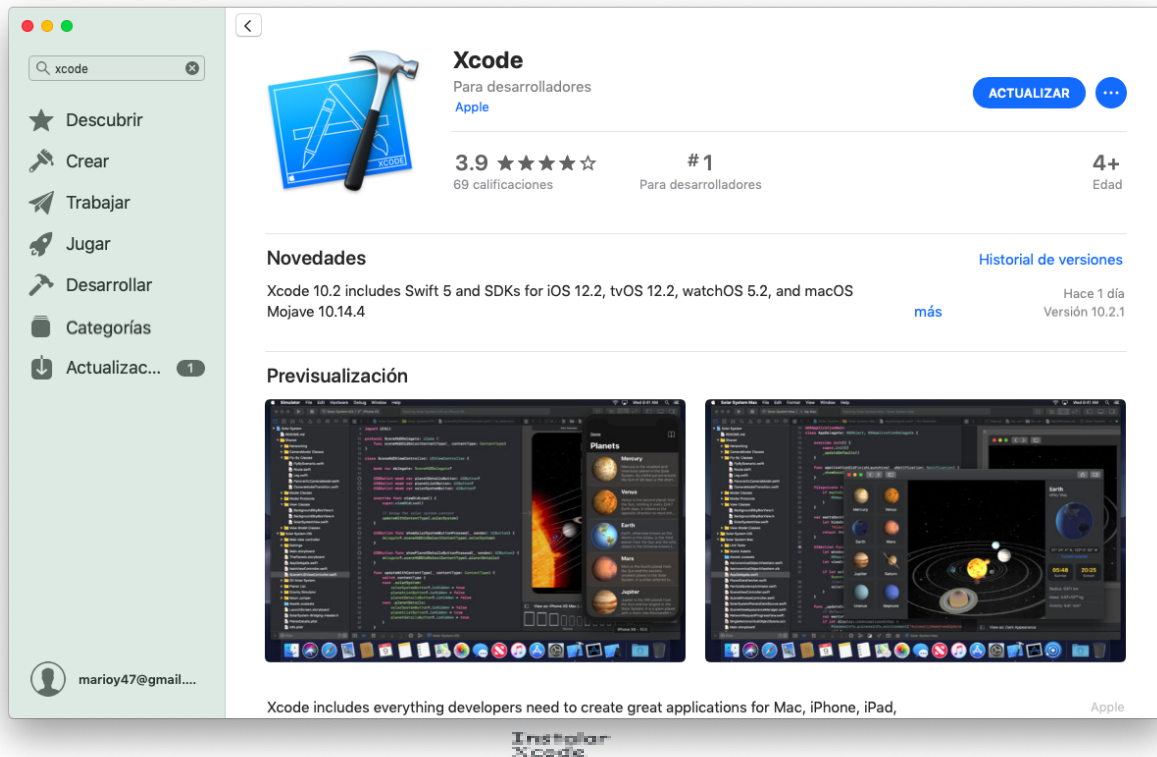
```
open -a "App Store"
```

... Luego busca Xcode ...



Buscar Xcode en el AppStore

Y por último haz click en **Instalar**



*En mi caso dice actualizar puesto que ya lo había instalado, pero el botón se ve igual. Esto puede tomar desde unos cuantos minutos hasta un par de horas mientras se descarga y se instala ya que la aplicación es bastante grande... Así que puedes ir a prepararte un café.*

## Configura los command-line tools

Ya que terminó de instalar Xcode, debemos instalar las librerías y comandos necesarios para poder desarrollar en flutter (y en go y en rust y en otros lenguajes de programación si así lo quieres).

Estos también son necesarios para que el comando brew, del cual hablaremos más adelante, funcione.

Para instalarlos en la terminal ejecuta el siguiente comando:

```
sudo xcode-select --switch  
/Applications/Xcode.app/Contents/Developer
```



Si abriste Xcode una primera vez, es posible que te haya pedido que aceptes la licencia de uso. Pero si no quieres abrirlo y hacerlo todo por la terminal (me gusta como piensas 😊), entonces debes ejecutar también el comando `xcodebuild`.

```
sudo xcodebuild -license
```

*Este comando requiere acceso de súper usuario. Es por esto que usamos `sudo`.*

*Debes usar la barra espaciadora para llegar hasta el final de texto que te aparecerá en la terminal. Y digitar `agree` para que la licencia quede instalada.*

## Instalar Flutter en Mac.

Llegamos a la parte interesante.

Vamos a descargar e instalar el comando `flutter` en Mac.

Es el comando que se encargará de compilar, verificar y lanzar los emuladores en tus proyectos con flutter, a tu computador.

El proceso será muy sencillo:

Primero que todo, debes descargarlo de <https://flutter.dev>.

*Mi directorio **home** es `/Users/Mario` y mi sub-directorio de proyectos es `Projects`, por lo que `cd ~/Projects` me llevará al directorio `/Users/Mario/Projects`. Adicionalmente mi directorio de descargas es `Downloads` por lo que `cd ~/Downloads` me llevará a `/Users/Mario/Downloads`*

## Descargar Flutter en Mac

Como este tutorial es para usar la línea de comandos, entonces lo vamos a descargar con `curl`

Luego en la terminal ejecuta curl de la siguiente manera.

```
cd ~/Downloads/  
curl https://storage.googleapis.com/flutter_infra/releases/stable/macos/flutter_macos_v1.2.1-stable
```

Aquí descargué flutter en la carpeta Downloads y le dí el nombre flutter.zip

Esta es una descarga de más de 400 megas. Así que puedes ir por un vaso de agua mientras tanto. Y deja el café para mañana y así no te desvelas.

# Descomprimir Flutter

Ahora que ya lo tienes en tu equipo, debes escoger un lugar en tu disco duro donde residirá de ahora en adelante.

Este lugar será fijo y no debe cambiar.

En mi caso escogí la ruta `/Users/Mario/Projects/flutter` como el lugar donde residirá de ahora en adelante.

Entonces debo ubicarme en esta carpeta y descomprimir el archivo .zip que descargué:

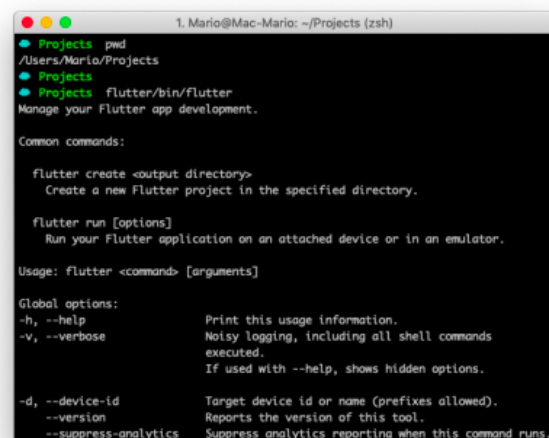
```
cd ~/Projects
unzip ~/Downloads/flutter.zip
```

Unzip creará la carpeta ~/Projects/flutter y guardará allí los archivos descargados.

Si todo va bien, puedes ejecutar el comando flutter de esta manera:

```
~/Projects/flutter/bin/flutter
```

Lo que te mostrará algo como esto:



Comando Flutter.

al contenido original en:  
io Code - GiancarloCode



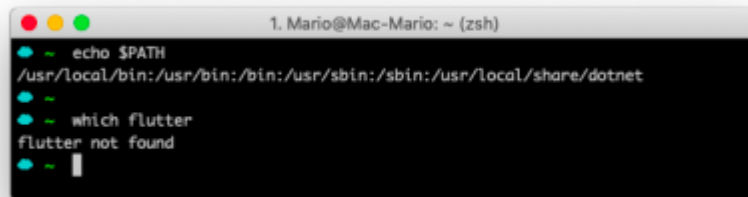
## Hacer que el comando flutter sea global

Es posible que desde ya estés fastidiado de tener que ejecutar  
~/Projects/flutter/bin/flutter (o /Users/Mario/Projects/flutter/bin/flutter)  
para ejecutar el comando.

Pero hasta el momento solo llevamos dos veces.

Para no tener que digitar esta ruta tan larga y simplemente usar flutter en la terminal, es necesario cambiar tu variable de entorno PATH

En términos sencillos, PATH guarda una lista de las diferentes ubicaciones en tu disco duro donde la terminal buscará comandos para ejecutar.



```
1. Mario@Mac-Mario: ~ (zsh)
~ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/share/dotnet
~ which flutter
flutter not found
```

Como te darás cuenta, mi PATH contiene las siguientes:

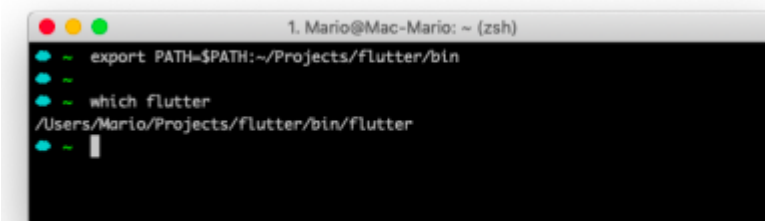
rutas /usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/share/dotnet, y no incluye /Users/Mario/Projects/flutter/bin. Es por esto que al ejecutar which flutter me responde que no encuentra el comando.

En PATH las rutas están separadas por :

Tenemos entonces que modificar esta variable de tal manera que reconozca ~/Projects/flutter/bin como un lugar con comandos.

Para esto se puede exportar y modificar la variable de esta manera.

```
export PATH=$PATH:~/Projects/flutter/bin
```



```
1. Mario@Mac-Mario: ~ (zsh)
~ export PATH=$PATH:~/Projects/flutter/bin
~ which flutter
/Users/Mario/Projects/flutter/bin/flutter
```





Ahora mi variable path tiene el siguiente

contenido/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/share/dotnet:/Users/Mario/Proyec

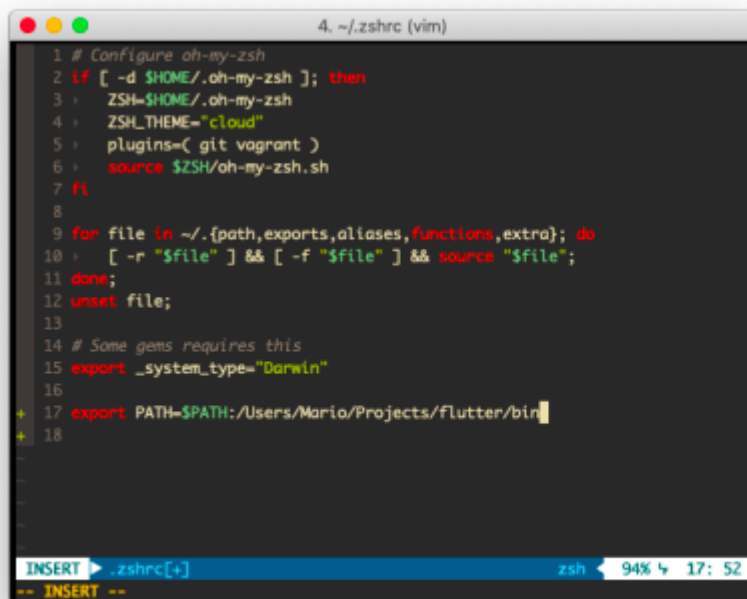
es lo que queríamos.

**Pero este cambio se perderá cuando cierre mi terminal!!!**

Si quiero hacer este cambio permanente, debo guardar en `.bashrc` (o `.zshrc` si utilizas `zsh`) este comando. La manera más sencilla de hacerlo es así:

```
echo "export PATH=\$PATH:~/Projects/flutter/bin" >> ~/.bashrc
```

También puedes editar el archivo directamente:



```
4. ~/.zshrc (vim)
1 # Configure oh-my-zsh
2 if [ -d $HOME/.oh-my-zsh ]; then
3   ZSH=$HOME/.oh-my-zsh
4   ZSH_THEME="cloud"
5   plugins=( git vagrant )
6   source $ZSH/oh-my-zsh.sh
7 fi
8
9 for file in ~/.{path,exports,aliases,functions,extra}; do
10   [ -r "$file" ] && [ -f "$file" ] && source "$file";
11 done;
12 unset file;
13
14 # Some gems requires this
15 export _system_type="Darwin"
16
17 export PATH=$PATH:~/Users/Mario/Projects/flutter/bin
18
```

Editando archivo `.bashrc`

Recuerda que en mi caso flutter se encuentra en `~/Projects/flutter/bin`

Con esto lo que hicimos fue agregar la nueva ruta a tu archivo de inicialización de la terminal, para que cada vez que la abras solo debas ejecutar flutter sin tener que hacer ningún tipo de export o escribir rutas raras.



## ¡Hagamos una prueba!

Cuando quieras instalar Flutter en Mac te recomiendo siempre hacer pruebas en el proceso.

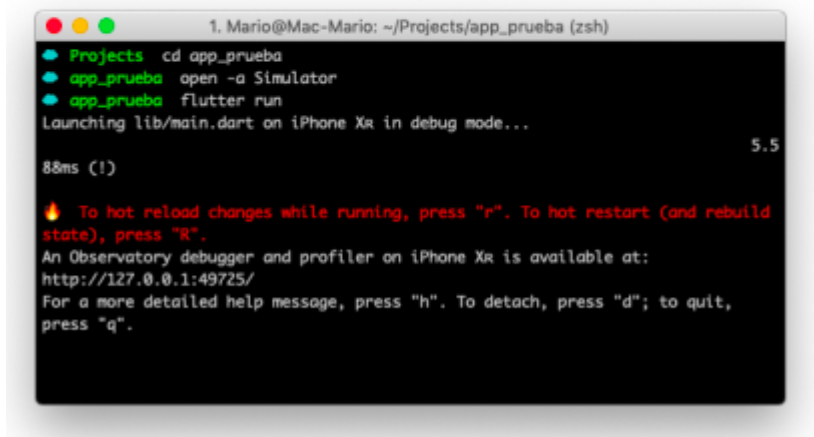
Aunque no hemos terminado. Si llegaste a este punto, puedes hacer aplicaciones en Flutter.

Lo único es que será algo difícil puesto que no hemos instalado buenas herramientas de desarrollo.

Pero hagamos una pausa y probemos que todo funciona en nuestro proceso para instalar Flutter en Mac.

Entonces ejecuta los siguientes pasos (ten paciencia, esto puede tomar un par de minutos por ser la primera vez):

```
cd ~/Projects # o donde guardes tus desarrollos
flutter create app_prueba
cd app_prueba
open -a Simulator
flutter run
```



### *App de prueba*



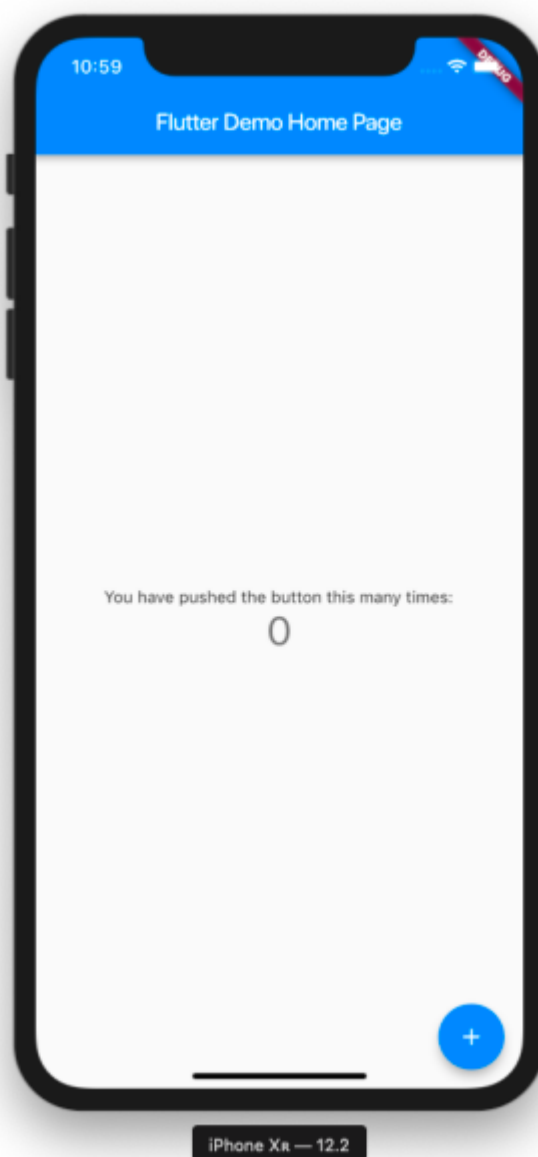
*Lo que estamos haciendo es creando una aplicación llamada `app_prueba`, luego abrimos el simulador de iPhone y por último ejecutamos esta aplicación en el simulador.*

*Si lo hiciste bien, debes ver en tu pantalla un simulador de iPhone*



App iniciando

*...con tu aplicación corriendo de la siguiente manera:*



***App corriendo***



## Instala brew.

*De aquí en adelante, estaremos usando mucho el comando brew el cual es bastante posible que ya conozcas.*

*Si no lo conoces entonces te invito a que visites <https://brew.sh> y aprendas sobre él.*

*En pocas palabras, es un comando que te permite descargar e instalar aplicaciones y paquetes de Mac con solo un comando.*

*Si has trabajado en Linux, piensa en brew como el apt-get (o yum) de MacOS.*

*Si no tenías brew instalado, entonces pega esto en un terminal y deja que tu computadora haga todo el trabajo.*

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

*Perfecto, de aquí en adelante vas a poder instalar el resto de paquetes desde la terminal.*

*... Aunque desafortunadamente la descarga e instalación de algunos componentes los vamos a tener que hacer de manera gráfica.*

*Siguiendo con el proceso de instalar Flutter en Mac...*

## Visual Studio Code.

No sé por qué, pero la selección de editor se ha convertido en una guerra santa.

Algunos usan vim otros usan Atom, otras personas juran que Sublime Text es el mesías y otras dicen que IntelliJ IDEA es lo último que vas a necesitar.

La verdad es que puedes desarrollar en flutter utilizando cualquier editor de texto incluyendo los que solo funcionan en la terminal como vim o emacs.

Sin embargo te recomiendo [Visual Studio Code](#) por cuatro razones principales:

Es relativamente liviano

- Tiene muy buen soporte a flutter
- Si programas en otros lenguajes como JavaScript, PHP o Ruby. Es mucho más fácil de trabajar usando el mismo editor
- Puedes extenderlo y configurarlo con cientos de extensiones que hay disponibles en el [marketplace](#).



Para instalar Visual Studio Code ejecutamos el siguiente comando en la terminal.

```
brew cask install visual-studio-code
```

```
1. Mario@Mac-Mario: ~ (zsh)
- brew cask install visual-studio-code
-> Satisfying dependencies
-> Downloading https://az764295.vo.msecnd.net/stable/51b0b28134d51361cf996d2f0a
##### 100.0%
-> Verifying SHA-256 checksum for Cask 'visual-studio-code'.
-> Installing Cask visual-studio-code
-> Moving App 'Visual Studio Code.app' to '/Applications/Visual Studio Code.app'
-> Linking Binary 'code' to '/usr/local/bin/code'.
visual-studio-code was successfully installed!
```

```
brew cask install visual-studio-code
```

## Extensiones de Visual Studio Code.

Ahora que tenemos Visual Studio Code instalado, vamos a instalar las extensión que nos permitirá programar con flutter... Pero lo vamos a hacer desde la terminal (Apuesto que no sabías que se podía)

Aunque solo necesitas la extensión de Flutter (la primera en la siguiente lista). Voy a agregar una serie de extensiones que me ayudan en el día a día como programador de flutter.

Entonces ejecutamos los siguientes comandos:

```
code --install-extension Dart-Code.flutter
code --install-extension fisheva.eva-theme
code --install-extension sleistner.vscode-fileutils
code --install-extension pkief.material-icon-theme
code --install-extension coenraads.bracket-pair-colorizer
```



```
2. Mario@Mac-Mario: ~ (zsh)
$ code --install-extension Dart-Code.flutter
code --install-extension fisheva.eva-theme
code --install-extension sleistner.vscode-fileutils
code --install-extension pkief.material-icon-theme
code --install-extension 2gua.rainbow-brackets
Found 'dart-code.flutter' in the marketplace.
Installing...
(node:93308) [DEP0005] DeprecationWarning: Buffer() is deprecated due to security and usability issues. Please use the Buffer.alloc(), Buffer.allocUnsafe(), or Buffer.from() methods instead.
Extension 'dart-code.flutter' v2.25.1 was successfully installed!
Found 'fisheva.eva-theme' in the marketplace.
Installing...
(node:93316) [DEP0005] DeprecationWarning: Buffer() is deprecated due to security and usability issues. Please use the Buffer.alloc(), Buffer.allocUnsafe(), or Buffer.from() methods instead.
Extension 'fisheva.eva-theme' v0.7.6 was successfully installed!
Found 'sleistner.vscode-fileutils' in the marketplace.
Installing...
(node:93325) [DEP0005] DeprecationWarning: Buffer() is deprecated due to security and usability issues. Please use the Buffer.alloc(), Buffer.allocUnsafe(), or Buffer.from() methods instead.
Extension 'sleistner.vscode-fileutils' v2.13.7 was successfully installed!
Found 'pkief.material-icon-theme' in the marketplace.
Installing...
(node:93333) [DEP0005] DeprecationWarning: Buffer() is deprecated due to security and usability issues. Please use the Buffer.alloc(), Buffer.allocUnsafe(), or Buffer.from() methods instead.
Extension 'pkief.material-icon-theme' v3.7.0 was successfully installed!
Found '2gua.rainbow-brackets' in the marketplace.
Installing...
(node:93341) [DEP0005] DeprecationWarning: Buffer() is deprecated due to security and usability issues. Please use the Buffer.alloc(), Buffer.allocUnsafe(), or Buffer.from() methods instead.
Extension '2gua.rainbow-brackets' v0.0.6 was successfully installed!
```

Instalando Extensiones de Visual

Te queda como tarea explorar qué hacen cada una de estas extensiones, ingresando a la sección de extensiones de Visual Studio Code y leyendo la introducción de cada una de ellas.

## Instalar Android Studio.

¿Creías que instalar Flutter en Mac era un par de clics? seguimos...

Hagamos un recuento de lo que hemos hecho:

- Instalamos **Xcode** y sus herramientas de desarrollo
- Instalamos **Flutter** y configuramos la terminal para poder usarlo



Esto quiere decir que hasta ahora tenemos configurada nuestra laptop para desarrollar aplicaciones de iOS cómodamente.

Pero si queremos desarrollar para Android (Cosa que yo quiero). Debemos instalar Android Studio y la herramienta de emulación de android.

Entonces ejecutemos brew para descargar e instalar Android Studio. Esto tomará un buen rato ya que es aproximadamente 1 Giga.

```
brew cask install android-studio
```

```
1. Mario@Mac-Mario: ~ (zsh)
~$ brew cask install android-studio
-> Satisfying dependencies
-> Downloading https://dl.google.com/dl/android/studio/install/3.4.0.18/android-studio-3.4.0.18.dmg 100.0%
-> Verifying SHA-256 checksum for Cask 'android-studio'.
-> Installing Cask android-studio
-> Moving App 'Android Studio.app' to '/Applications/Android Studio.app'.
android-studio was successfully installed!
```

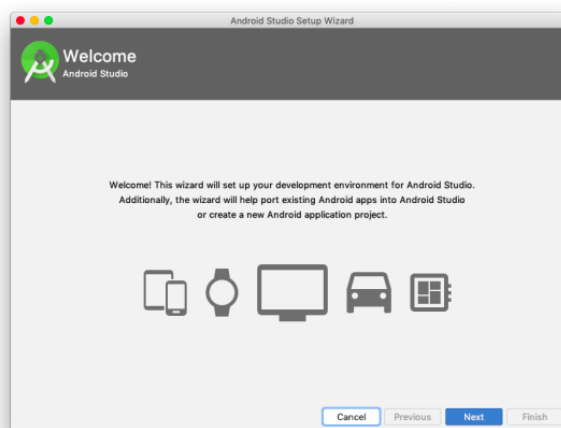
brew cask install android-studio

## Primer inicio e Instalar un SDK

De Android Studio solo usaremos su emulador. Así que vamos a iniciarlo, pero no vamos a configurarlo.

Desafortunadamente esto se debe hacer en la interfaz gráfica.

Entonces, arráncalo y haz clic en cancel en la primera ventana del wizard de configuración.



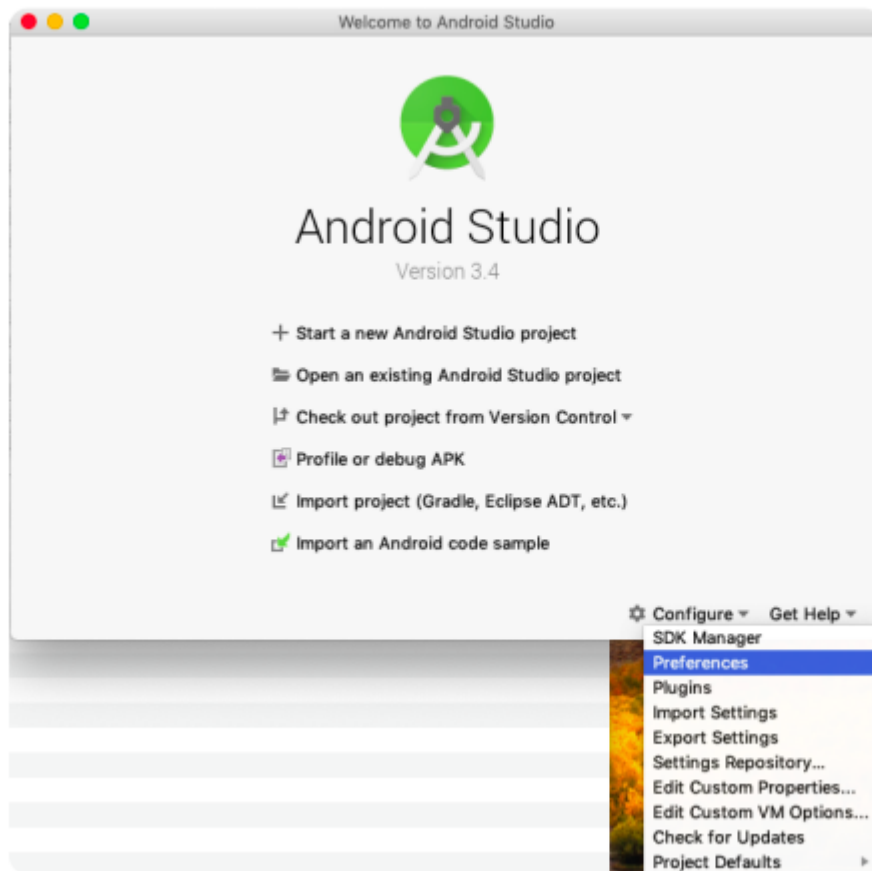
**Cancelar el wizard de configuración**





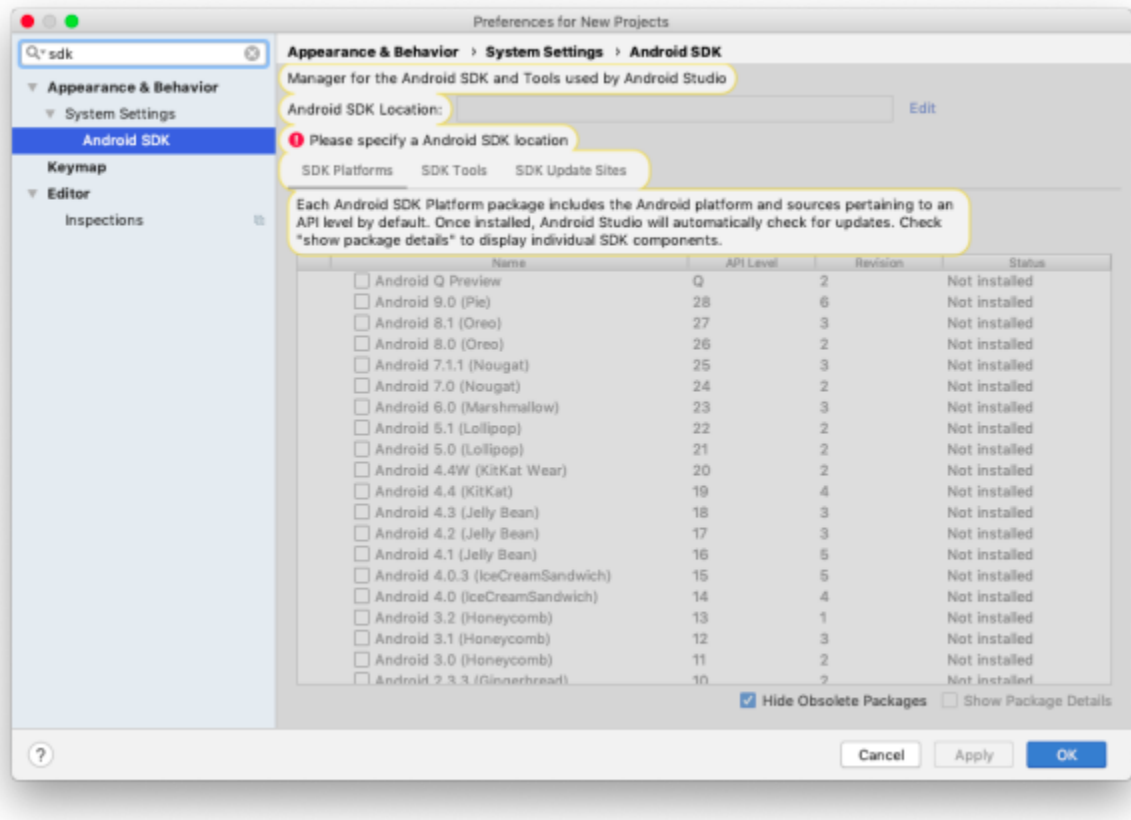
## Instalar el SDK de Android.

El SDK es necesario para poder compilar aplicaciones para Android. Entonces, en la ventana de Bienvenida, selecciona “preferences”...



Android Studio Preferences

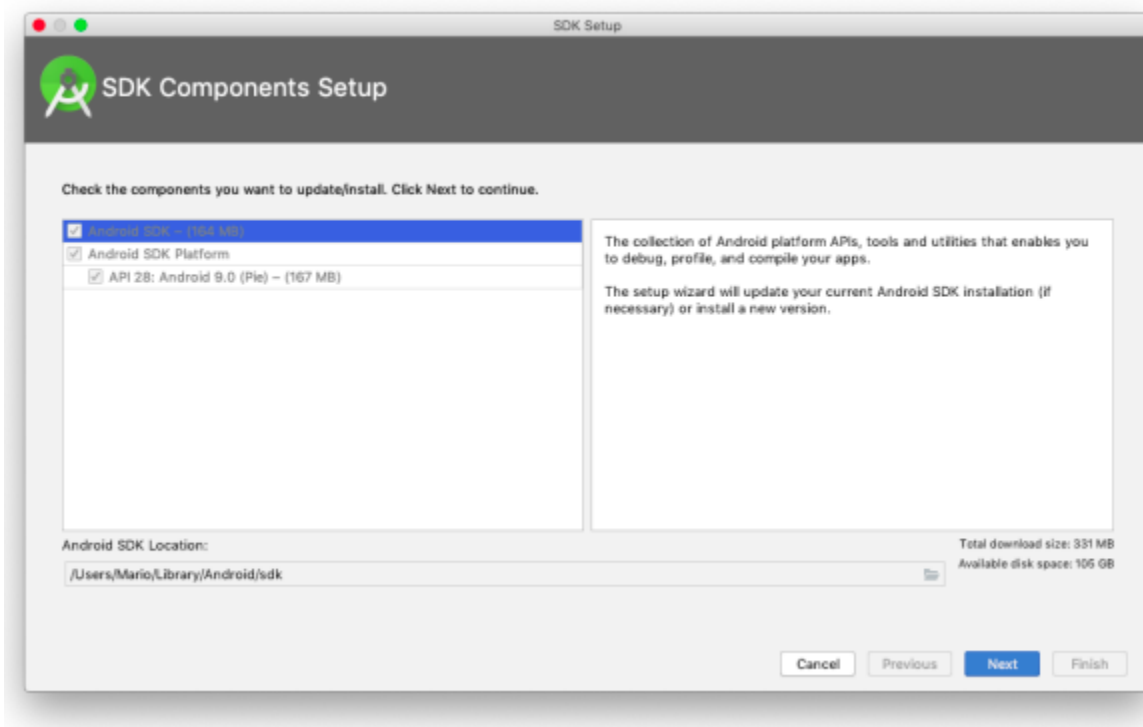
***Y en las preferencias busca la configuración del Android SDK***



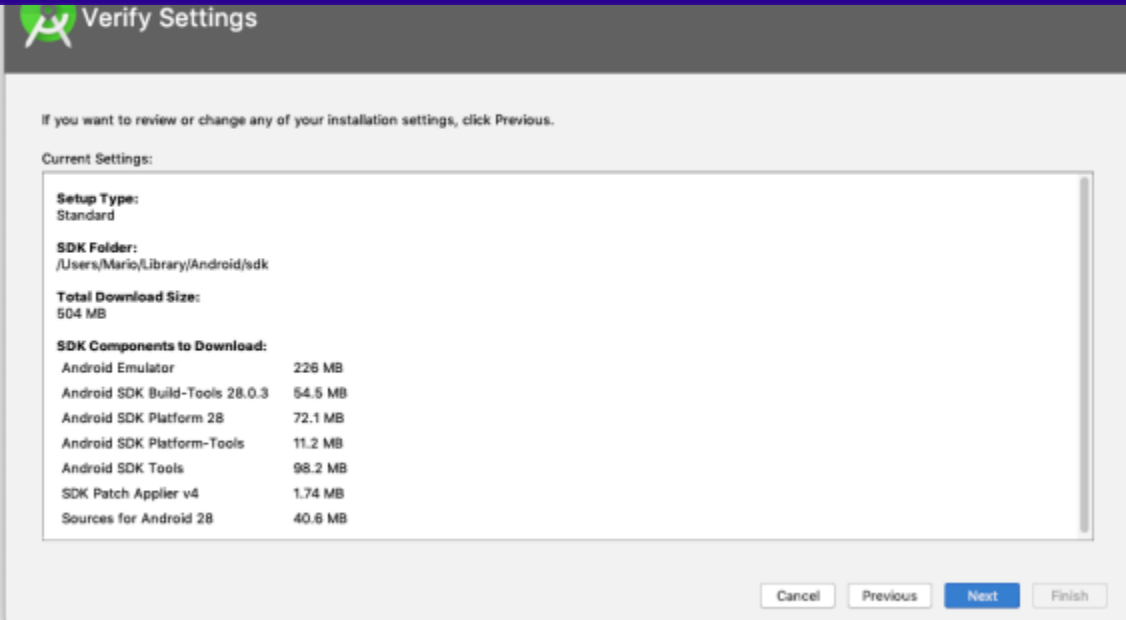
## Buscar Preferencias Android SDK

Has click en **edit** para que empieces un nuevo wizard de instalación del SDK.

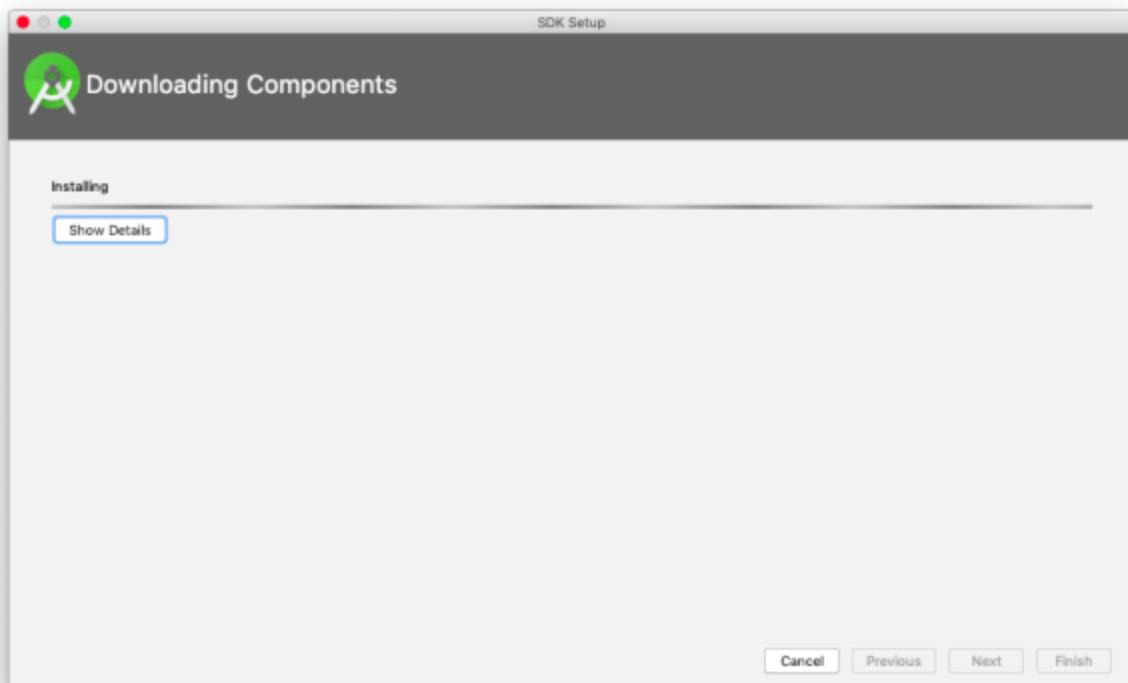
Esto te mostrará un ventana de instalación de SDK



## ***Instalación de SDK***



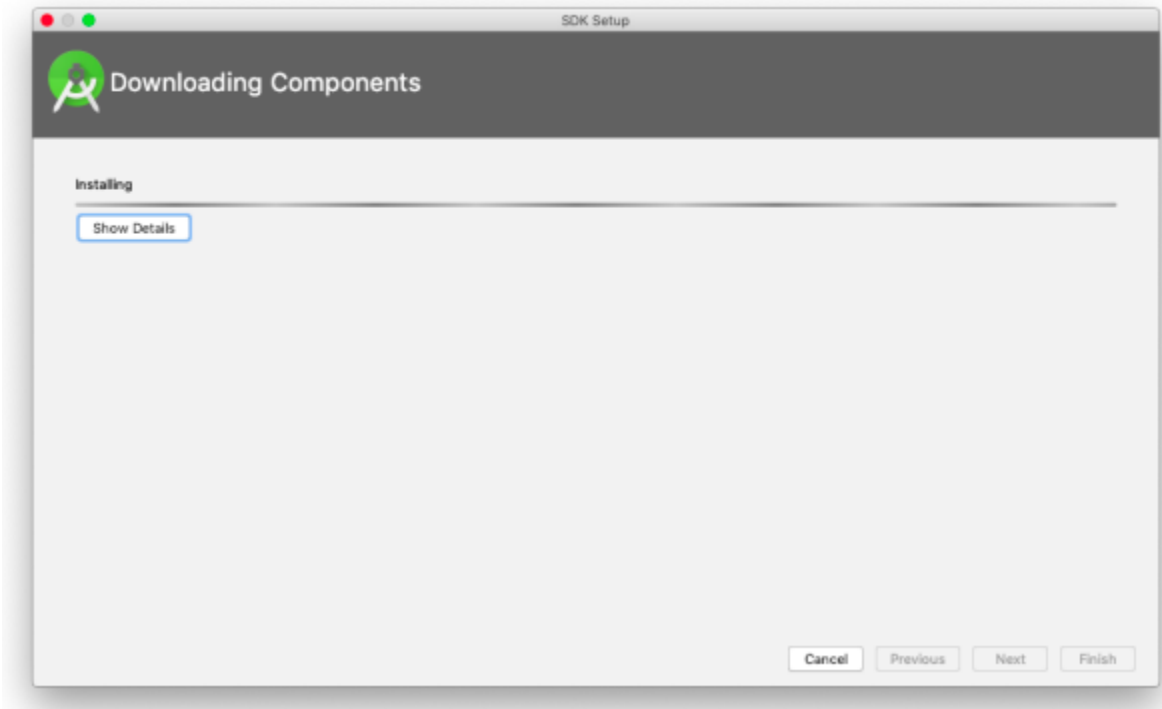
## *Instalación de SDK – Resumen*



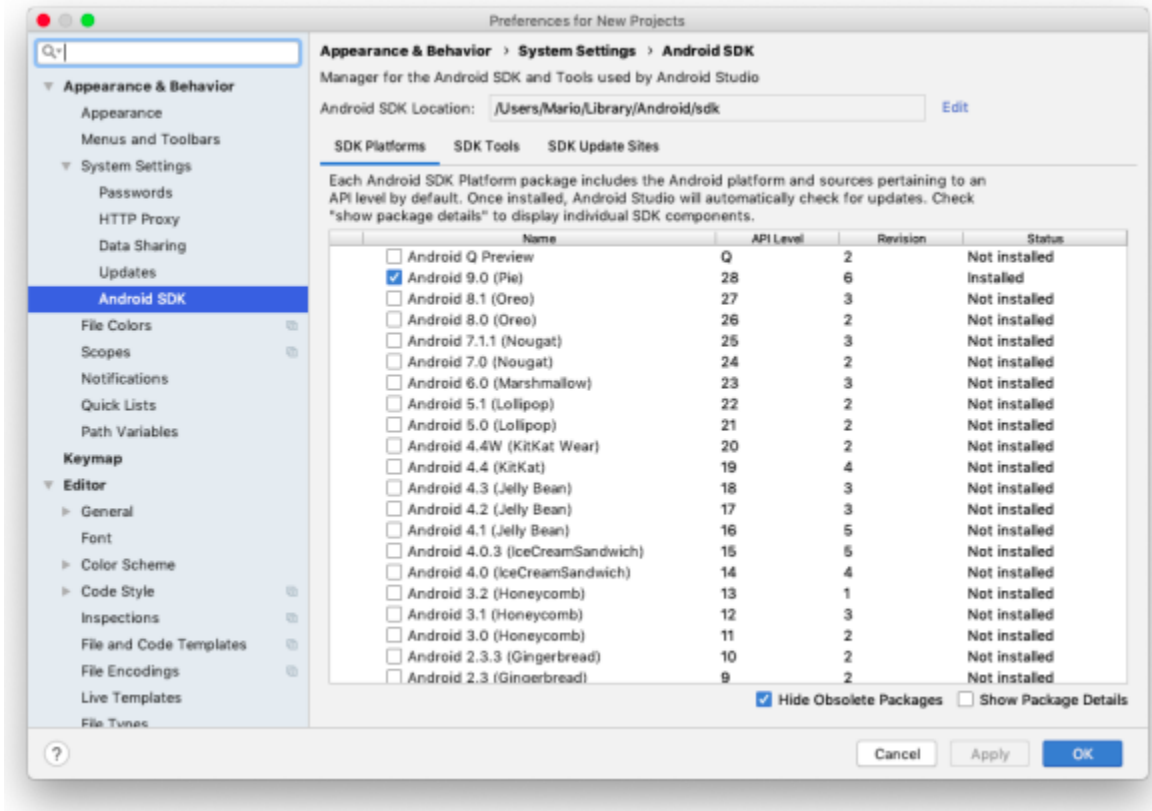
## *Instalación de SDK – Progreso*



*pantalla de preferencias, pero con un SDK instalado (en este caso Android 9.0 Pie)*



***Instalación de SDK – Progreso***



*Resultado de instalación de SDK*

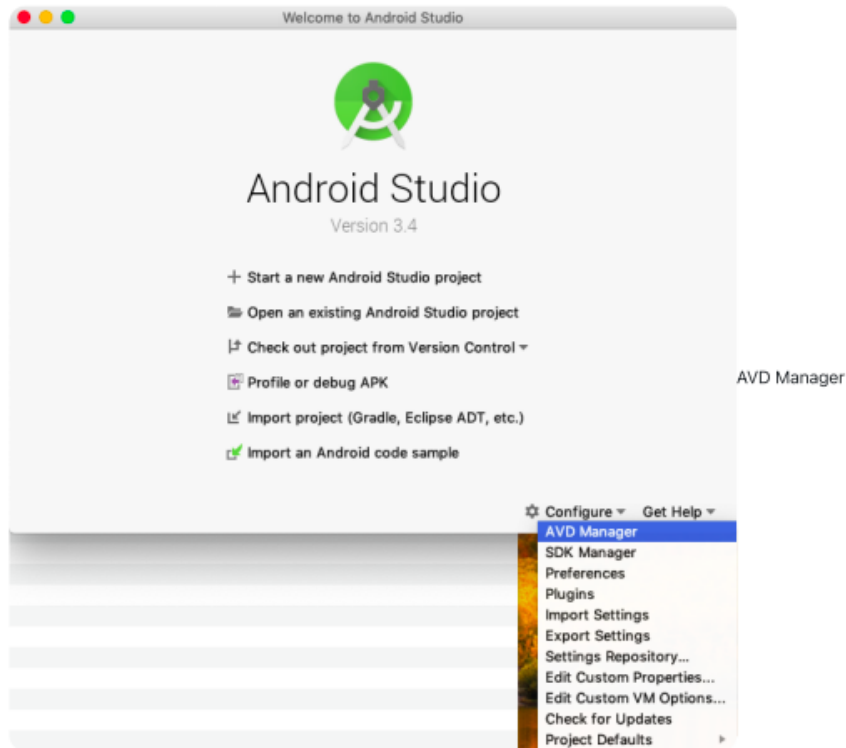
## Configurar un emulador de Android

Tengo que admitirlo...

Desarrollar para Android puede ser un dolor de cabeza por todas las dependencias y requisitos que debes cumplir. No solo tienes que descargar un editor y un SDK, sino que también debes **descargar** y **configurar** un dispositivo de emulación.

Entonces vamos a hacer la descarga desde Android Studio y la configuración desde la terminal.

Para la descarga, vuelve a la pantalla de bienvenida de Android Studio y selecciona **AVD Manager**.

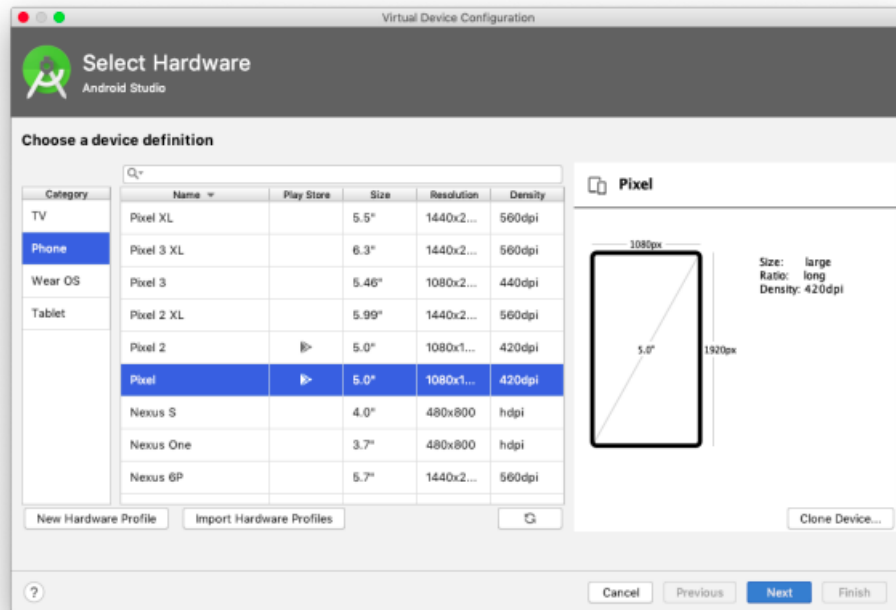


***Luego***



***Create Virtual Device***

***Y después selecciona un dispositivo.***

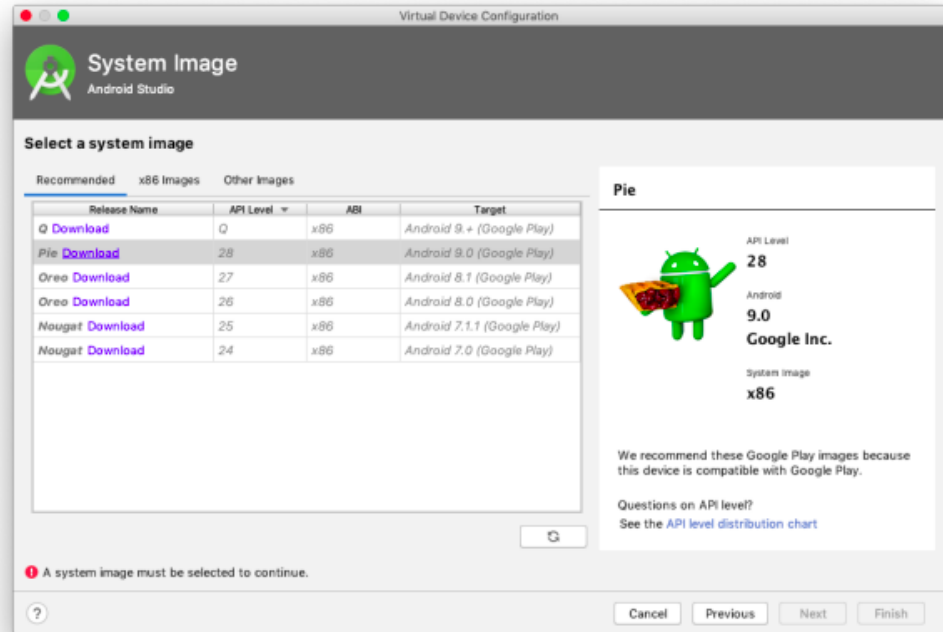


### ***Seleccionar tipo de Virtual Device***

***En este momento no importa cuál vas a crear, puesto que solo instalaremos los archivos de emulación, más no terminaremos la configuración del emulador en sí. Esa parte la haremos en la terminal.***

***Luego selecciona **Download** en la columna de Release Name***

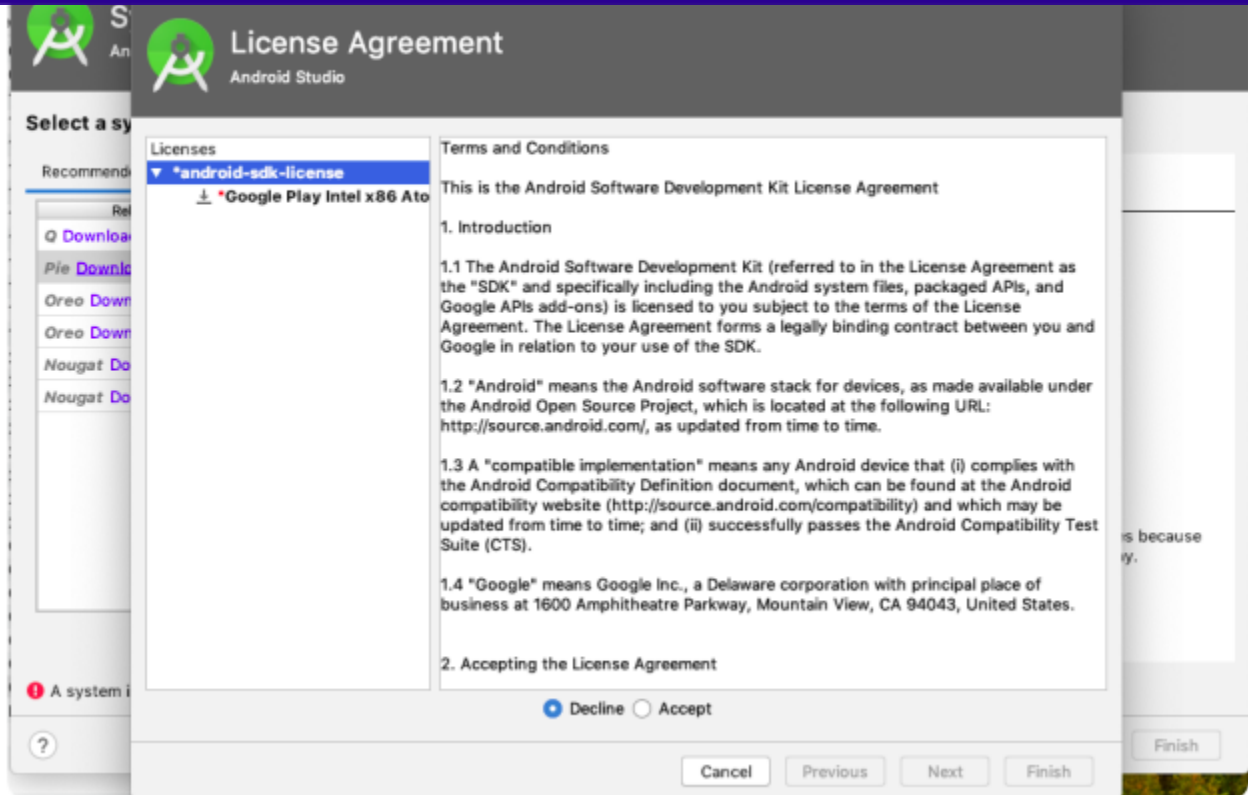




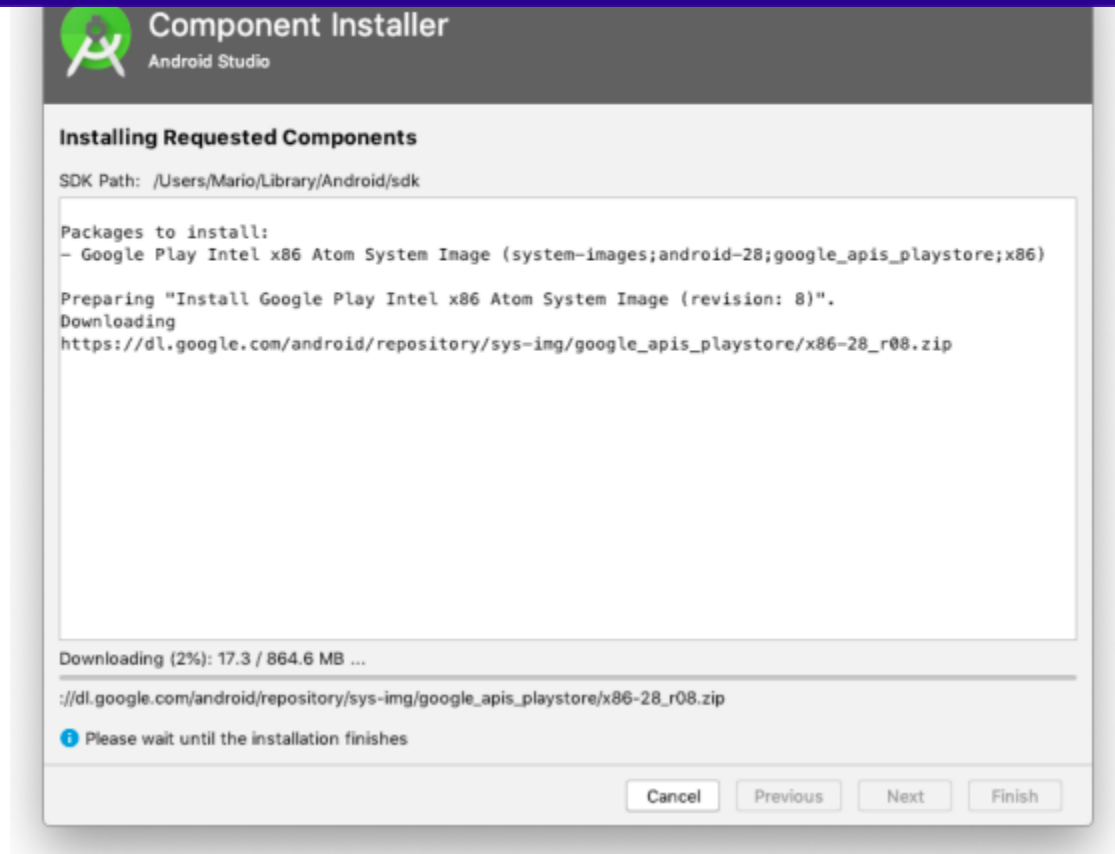
## Seleccionar tipo de Virtual Device

*Yo escogí Pie puesto que al momento de escribir este artículo Q estaba todavía en modo Preview y no quería arriesgarme a tener una instalación incompleta.*

*Haz clic en **next** en las próximas **DOS** siguientes ventanas:*

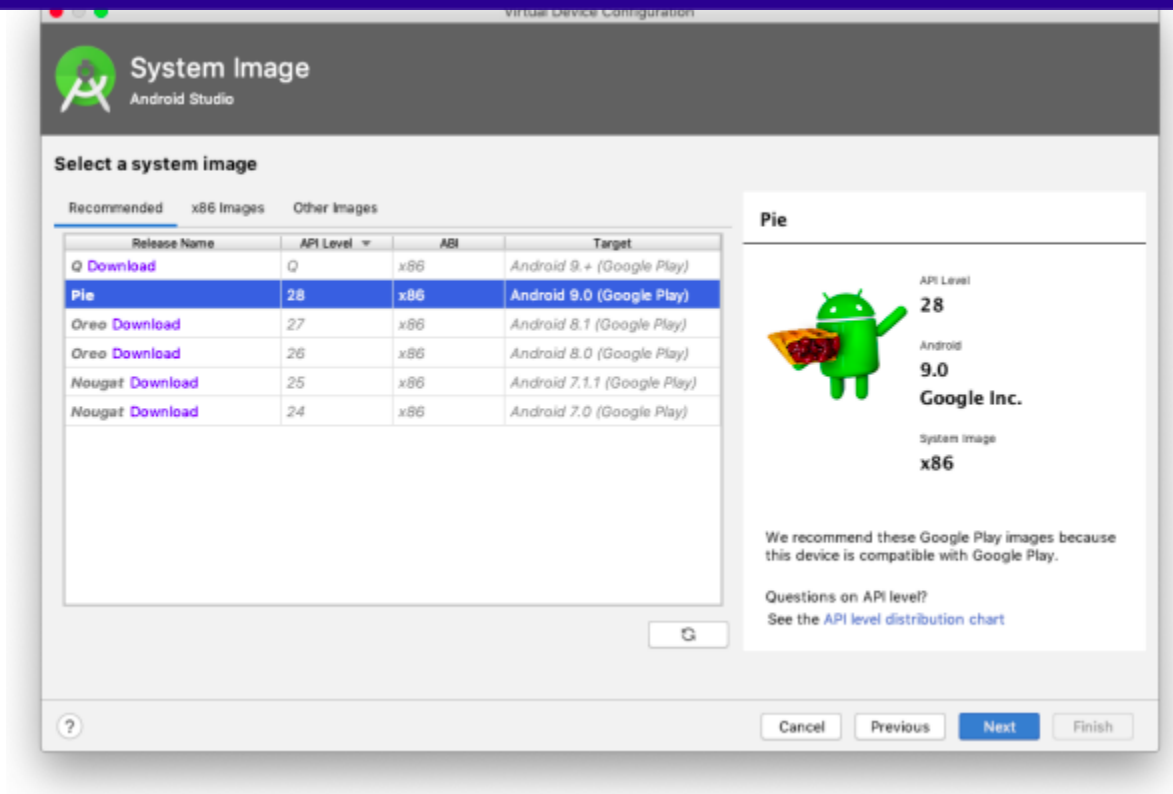


### *Info legal de instalar imagen*



***Progreso de instalación de imagen***

***En la tercera ventana haz clic en Cancel. La razón es porque vamos a realizar la creación de la imagen en la terminal.***



*Cancelar la instalación del dispositivo*

No hay nada de malo en hacer click en next aquí... es solo que el ejercicio que estamos haciendo en este artículo es de hacer la mayor cantidad de pasos en la terminal.

**### Crear el emulador de Android** Ya que tenemos el SDK y una imagen descargada. Podemos crear el emulador con flutter.

**Vuelve a la terminal y ejecuta**

```
flutter emulators --create --name pixel
```



```
1. Mario@Mac-Mario: ~ (zsh)
$ flutter emulators
1 available emulator:

apple_ios_simulator • iOS Simulator • Apple

To run an emulator, run 'flutter emulators --launch <emulator id>'.
To create a new emulator, run 'flutter emulators --create [--name xyz]'.

You can find more information on managing emulators at the links below:
https://developer.android.com/studio/run/managing-avds
https://developer.android.com/studio/command-line/avdmanager

$ flutter emulators --create --name pixel
Emulator 'pixel' created successfully.

$ flutter emulators
2 available emulators:

pixel • pixel • Google
apple_ios_simulator • iOS Simulator • Apple

To run an emulator, run 'flutter emulators --launch <emulator id>'.
To create a new emulator, run 'flutter emulators --create [--name xyz]'.

You can find more information on managing emulators at the links below:
https://developer.android.com/studio/run/managing-avds
https://developer.android.com/studio/command-line/avdmanager

$
```

***Crear un nuevo emulador con flutter en mac***

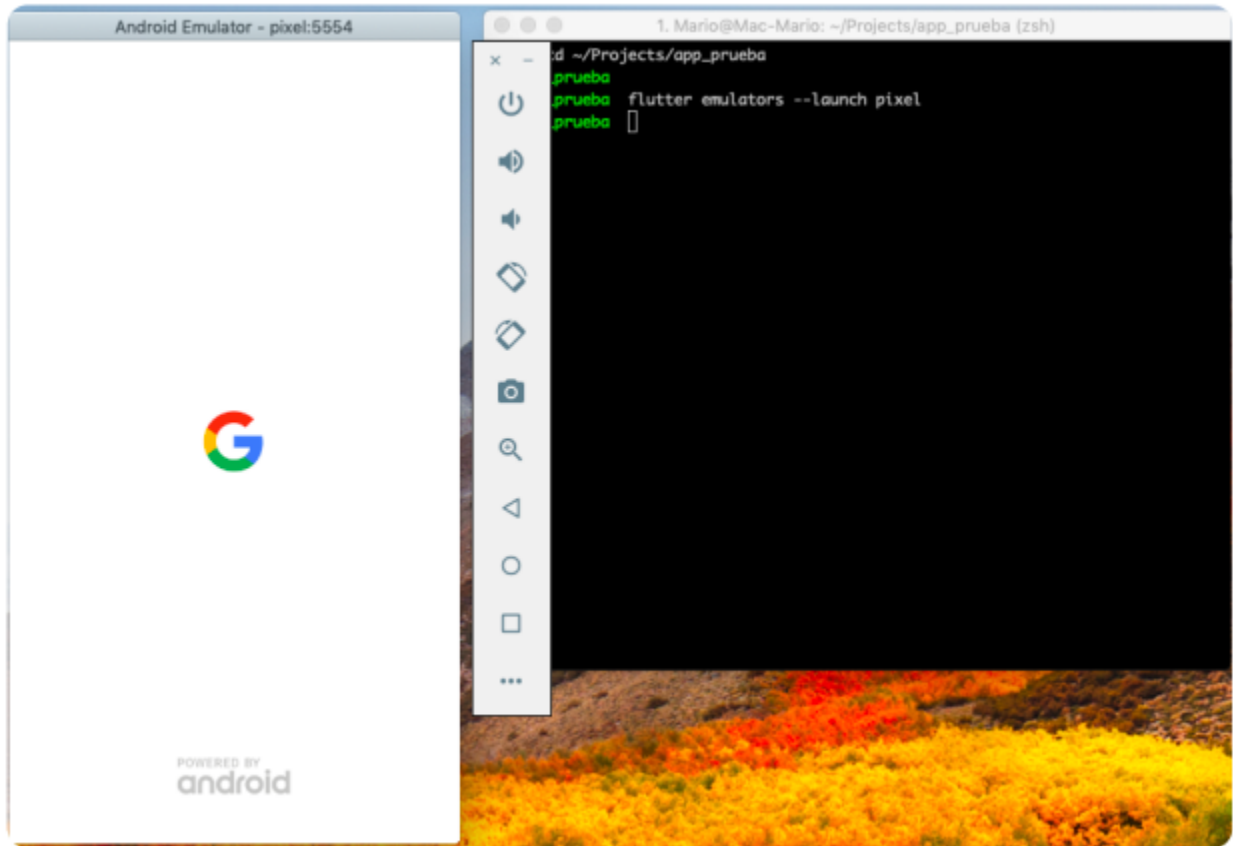
***En este caso pixel es el nombre que le dí a mi emulador con el fin de memorizarlo fácilmente***

***Si llegaste hasta aquí, FELICITACIONES... ya puedes probar tus aplicaciones en Android.***

***Para probar, ejecuta.***

```
flutter emulators --launch pixel
```

***Y ten mucha paciencia, pues esta primera carga puede tomar mucho tiempo.***



## *Lanzando el emulador de Android*

*Y por último, último, último, puedes correr la aplicación de prueba en tu nuevo emulador con*

```
flutter run
```